An Axiomatic Approach to Measure the Degree of Dirtiness in Relational Databases [†]

Maria Vanina Martinez
mvm@cs.umd.edu

Department of Computer Science
University of Maryland College Park

Abstract

There has been a significant amount of interest in recent years on how to reason about inconsistent knowledge bases. However, with the exception of three papers by Lozinskii, Hunter and Konieczny and by Grant and Hunter, there has been almost no work on characterizing the degree of dirtiness of a database. One can conceive of many reasonable ways of characterizing how dirty a database is. Rather than choose one of many possible measures, we present a set of axioms that any dirtiness measure must satisfy. We then present several plausible candidate dirtiness measures from the literature (including those of Hunter-Konieczny and Grant-Hunter) and identify which of these satisfy our axioms and which do not. Moreover, we define a new dirtiness measure which satisfies all of our axioms.

1 Introduction

It is an open secret that most commercial databases are *dirty* and in fact, there is a wide range of companies (e.g. SAS, Ascential – previously known as Informix) that offer data cleaning services. However, to date, with the exception of some ground-breaking work [Loz94, HK05, GH06], we are not aware of any work that attempts to actually characterize how dirty a database is, and thus there is no objective measure to assess whether an allegedly cleansed database is in fact significantly cleaner than the original.

In this paper, we focus on a more restricted scenario than [Loz94, HK05, GH06]. We focus on inconsistency in just *relational* databases (i.e. tables of tuples) with associated functional dependencies [Ull89] that form one of the most important types of integrity constraints used in databases. Intuitively, functional dependencies say that when certain attribute values are equal, then other attribute values must be equal as well. A good example of a functional dependency is one which says that in the same database, each person's salary is unique. We also assume the existence of a total order on attributes (i.e. columns) in the relational table that indicates how "reliable" those attributes are. Thus, in an employee database, we may choose to believe that the social security number attribute is more reliable than the salary attribute. However, unlike Hunter-Konieczny's and Grant-Hunter's work which is primarily symbolic, our work is geared to inconsistency in numerical data – this is critical in real world

[†]Submitted to the Department of Computer Science, University of Maryland College Park, in partial fulfilment of the requirements for degree of Master in Science in Computer Science. The work described here is based on work done in collaboration with V.S. Subrahmanian, Henri Prade, Gerardo I. Simari, and Andrea Pugliese [MPS+07].

databases as they almost always contain numeric data such as salary data, sales figures, shipping costs, etc.

We then propose a general set of three axioms that we believe *any* measure of database dirtiness must satisfy when a *single* functional dependency is present. We subsequently assess several dirtiness measures, both naive and from the past (including the Hunter-Konieczny and Grant-Hunter measures) to see how they comply with the axioms and show that they all fail to satisfy all the axioms. We finally define a new dirtiness measure which satisfies all the axioms. Subsequently, we show how any such single FD dirtiness measure can be used to build a dirtiness measure that handles multiple functional dependencies and we present a couple of example measures of this type.

2 Syntax and Notation

We assume the existence of a relational schema $R = (A_1, \ldots, A_n)$ [Ull89] where the A_i's are attributes. Each attribute A_i has an associated domain, $dom(A_i)$. A *tuple* over R is a member of $dom(A_1) \times \cdots \times dom(A_n)$. A *database* DB is any finite set of tuples over R. In the rest of this paper, we assume R is arbitrary but fixed. For example, Fig. 1 shows a database over the schema (Name, Age, Height) with the obvious domains. Each row in the figure is a tuple. We assume the existence of a set of symbols called *tuple variables* that range over the tuples in DB. A *functional dependency* for database DB is any expression of the form $\forall t, t' \in DB,\ t.A_{i_1} = t'.A_{i_1} \wedge \ldots \wedge t.A_{i_t} = t'.A_{i_t} \Rightarrow t.A_{i_{t+1}} = t'.A_{i_{t+1}} \wedge \ldots \wedge t.A_{i_m} = t'.A_{i_m}$. For instance, in Fig. 1, $t.Name = t'.Name \Rightarrow t.Age = t'.Age$ is an example of a functional dependency saying that two tuples about the same person should agree on age.

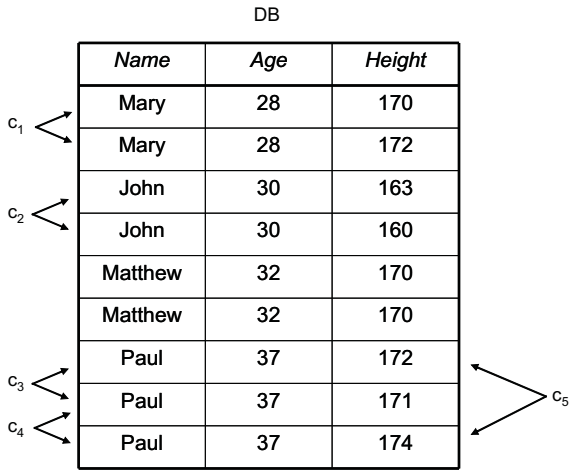

Figure 1: An example database

Without loss of generality, we assume two functional dependencies cannot have the same antecedent. We assume that the attributes in a table are totally ordered by a *reliability ordering* $>_r$ (that can be derived, e.g., from inherent properties of attribute domains or from historical error statistics). We write $A_i >_r A_j$ iff attribute A_i is more reliable than attribute A_j.

3 Culprits, Clusters, and Dirtiness Functions

Our notion of database dirtiness is based on the concepts of culprits and clusters. Culprits are just the duals of maximal consistent subsets which have been widely studied [Loz94, HK05, GH06, BKMS92]. Clusters, on the other hand, do not seem to have been studied much in AI. Both of these parameters will be used in our axiomatic characterization of the dirtiness of a database.

Definition 3.1 *Let DB be a database and $\mathcal{F}$ a set of functional dependencies. A* culprit *is a set $c \subseteq DB$ not satisfying $\mathcal{F}$ such that $\forall\, c' \subset c$, c' satisfies $\mathcal{F}$.*

Thus, culprits are minimal sets of database tuples that cause a functional dependency violation. Let $culprits(DB, \mathcal{F})$ denote the set of culprits in DB w.r.t. $\mathcal{F}$.

Example 3.1 *Consider a functional dependency fd stating that $\forall t, t' \in DB$, $t.Name = t'.Name \Rightarrow t.Age = t'.Age \wedge t.Height = t'.Height$. The relation in Fig. 1 has five culprits w.r.t. fd, denoted by c_1, c_2, c_3, c_4, c_5.*

The following proposition states that the $culprits(DB, \mathcal{F})$ function is monotonic w.r.t. DB.

Proposition 3.1 *If $DB' \subseteq DB$, then $culprits(DB', \mathcal{F}) \subseteq culprits(DB, \mathcal{F})$.*

Definition 3.2 *Let DB be a database and $\mathcal{F}$ a set of functional dependencies. Given two culprits $c, c' \in culprits(DB, \mathcal{F})$, we say that c and c'* overlap*, denoted $c \triangle c'$, iff $c \cap c' \neq \emptyset$.*

Definition 3.3 *Let $\triangle^*$ be the reflexive transitive closure of relation $\triangle$. A* cluster *is a set $cl = \bigcup_{c \in e} c$ where e is an equivalence class of $\triangle^*$.*

We denote with $clusters(DB, \mathcal{F})$ the set of all clusters in DB w.r.t. $\mathcal{F}$. We now present an example of overlapping culprits and clusters.

Example 3.2 *In Fig. 1, the pairs of overlapping culprits in database DB are (c_1, c_1), (c_2, c_2), (c_3, c_3), (c_4, c_4), (c_5, c_5), (c_3, c_4), (c_3, c_5), (c_4, c_5), and all of the symmetric pairs. Therefore, the clusters in DB are the sets $cl_1 = \{(Mary, 28, 170), (Mary, 28, 172)\}$, $cl_2 = \{(John, 30, 163), (John, 30, 160)\}$, and $cl_3 = \{(Paul, 37, 172), (Paul, 37, 171), (Paul, 37, 174)\}$.*

Clusters are important because they localize the inconsistencies. For instance, clusters cl_1, cl_2, cl_3 above tell us that there is something wrong with the *Mary, John* and *Paul* triples respectively.

We now define single-dependency and multiple-dependency dirtiness functions.

Definition 3.4 *A* single-dependency (resp. multiple-dependency) dirtiness function δ *takes a database instance DB, a functional dependency fd (resp. a finite set $\mathcal{F}$ of functional dependencies), and a reliability ordering $>_r$ and returns as output a real number in the left-closed, right-open interval $[0, \infty)$.*

4 Axioms

Our first axiom on single-dependency dirtiness functions δ says that consistent databases have a dirtiness level of 0.

Axiom S1. If $culprits(DB, \{fd\}) = \emptyset$, then $\delta(DB, fd, >_r) = 0$.

Our second axiom is based on the statistical notions of standard deviation and variance (which is the square of s.d.), which have been used for decades by the statistics community as a measure of dirtiness in a data set, to define an axiom dirtiness functions should satisfy.

We first generalize the notion of variance to string attributes. Given a numeric attribute A, let $variance_A : 2^{dom(A)} \rightarrow \mathbb{R}^+$ be the variance of A. When $dom(A)$ is a set of strings, $variance_A$ builds on top of string similarity-evaluation function (e.g. edit distance, Hamming distance, Levenshtein distance). Given a set of strings S and a similarity-evaluation function $sim : string \times string \rightarrow \mathbb{R}^+$, let s_{min} be the first string appearing in S according to lexicographic order. The $variance_A(S)$ function returns the variance of the set $D = \{sim(s_{min}, s) \mid s \in S\}$.

From now on, the sequence of attributes in a functional dependency fd, ordered w.r.t. $>_r$, is denoted $\{A_{fd,1}, \ldots, A_{fd,m}\}$. Thus, $A_{fd,1}$ is the most reliable attribute in fd, $A_{fd,2}$ is the second most reliable attribute in fd, and so forth.

Definition 4.1 *Let fd be a functional dependency, and cl, cl' be two clusters. We say that $cl' \sqsubseteq^{fd}_{var} cl$, read "$cl'$ is less or equally varied than cl w.r.t. fd" iff $\exists j \in [1, m]$ s.t. $variance_{A_{fd,j}}(cl'.A_{fd,j}) \leq variance_{A_{fd,j}}(cl.A_{fd,j})$, and $\forall i < j$, $variance_{A_{fd,i}}(cl'.A_{fd,i}) = variance_{A_{fd,i}}(cl.A_{fd,i})$. We also say that $cl' \sqsubset^{fd}_{var} cl$, read "$cl'$ is less varied than cl w.r.t. fd" iff $\exists j \in [1, m]$ s.t. $variance_{A_{fd,j}}(cl'.A_{fd,j}) < variance_{A_{fd,j}}(cl.A_{fd,j})$, and $\forall i < j$, $variance_{A_{fd,i}}(cl'.A_{fd,i}) = variance_{A_{fd,i}}(cl.A_{fd,i})$.*

The above definition says that cl' is less or equally varied than cl w.r.t. fd iff as we examine the attribute in fd in decreasing order of reliability, the first attribute on which they have differing variances is one where cl' has a lower variance than cl.

Definition 4.2 *We say that DB'* is preferable to *DB* w.r.t. the dependency *fd, denoted $DB' \succ_{fd} DB$, iff there exists a function*

$$\alpha : clusters(DB', \{fd\}) \rightarrow clusters(DB, \{fd\})$$

such that $\forall cl' \in clusters(DB', \{fd\})$ it holds that:

- *$cl' \sqsubseteq^{fd}_{var} \alpha(cl')$;*
- *cl' and $\alpha(cl')$ agree on all attributes that appear in the body of fd;*

and at least one of the following conditions holds:

- *$\exists cl' \in clusters(DB', \{fd\})$ such that $cl' \sqsubset^{fd}_{var} \alpha(cl')$;*

- $\exists cl \in clusters(DB, \{fd\})$ *such that* $\nexists cl' \in clusters(DB', \{fd\})$, $\alpha(cl') = cl$.

Intuitively, DB' is preferable to DB with respect to variance if there is a mapping between the clusters of DB' and the clusters of DB such that (i) each of the clusters in DB' shows less or equal variance than its image; (ii) either there exists a cluster in DB' having strictly less variance than its image in DB, or there exists a cluster in DB that does not belong to the codomain of the mapping.[1] This definition leads us directly to:

Axiom S2. If $DB' \succ_{fd} DB$, then $\delta(DB', fd, >_r) < \delta(DB, fd, >_r)$.

Example 4.1 *Consider the databases in Fig. 2. Cluster* cl_4 *shows lower variance than* cl_1*; Clusters* cl_2 *and* cl_5 *are equal; Cluster* cl_6 *shows lower variance than* cl_3*. Therefore, Axiom S2 dictates that* DB' *has a lower dirtiness degree than* DB*.*

DB

Name	*Age*	*Height*	
Mary	30	170	cl_1
Mary	30	171	
Mary	30	172	
Matthew	32	153	
John	32	163	cl_2
John	32	160	
Matthew	32	153	
Paul	35	172	cl_3
Paul	35	171	
Paul	35	174	

DB'

Name	Age	*Height*	
Mary	30	170	cl_4
Mary	30	170.5	
Mary	30	171	
Charles	35	169	
John	32	163	cl_5
John	32	160	
Matthew	32	175	
Paul	35	172	cl_6
Paul	35	171.5	
Paul	35	171	

Figure 2: According to Axiom S2, DB' has a lower dirtiness degree than DB

We also consider a weaker variant of Axiom S2 called S2′:

Axiom S2′. If $DB' \succ_{fd} DB$ and $\forall cl' \in clusters(DB', \{fd\})$ $cl' \subseteq \alpha(cl')$, then $\delta(DB', fd, >_r) < \delta(DB, fd, >_r)$.

The condition that $\forall cl' \in clusters(DB', \{fd\})$ $cl' \subseteq \alpha(cl')$ is not satisfied by the databases in Fig. 2. Hence, Axiom S2′ does not impose any restrictions on δ. However, if (Mary, 30, 170.5) and (Paul, 35, 171.5) were not present in DB', then Axiom S2′ would instead require $\delta(DB', fd, >_r) < \delta(DB, fd, >_r)$.

5 Examples of Single-Dependency Dirtiness Functions

In this section, we present some single-dependency dirtiness functions.

[1] It has been argued that when the values of disagreeing attributes are too far apart, they should simply be considered inconciliable [BDP+02]. In our case the objective is that of assessing the degree of dirtiness, so we still look at variances.

5.1 Naive Culprits-based Single-Dependency Dirtiness Functions

The following two simple dirtiness functions are based on culprits:

1. $|culprits(DB, \{fd\})|$
2. $\sum_{c \in culprits(DB,\{fd\})} |c|$

The first measure above just counts the number of culprits, the second sums up the number of tuples in each culprit.

Proposition 5.1 *The naive culprits-based dirtiness functions satisfy Axioms S1 and S2′.*

It is easy to see that these two measures, both of which seem reasonable at first sight, *do not satisfy Axiom S2.* To see why, consider the databases in Fig. 2. Here, we have $|culprits(DB, \{fd\})| = |culprits(DB', \{fd\})|$ and $\sum_{c \in culprits(DB,\{fd\})} |c| = \sum_{c \in culprits(DB',\{fd\})} |c|$, whereas Axiom S2 states that DB' should have a lower dirtiness degree.

5.2 Naive Cluster-based Single-Dependency Dirtiness Functions

We now define two cluster-based dirtiness functions:

1. $|clusters(DB, \{fd\})|$
2. $\sum_{cl \in clusters(DB,\{fd\})} |cl|$

As in the case of the culprit based dirtiness functions, the first measure simply counts the number of clusters, while the second counts the sum of the number of tuples in each cluster.

Proposition 5.2 *Dirtiness function 1 above satisfies Axiom S1.*

It is easy to see that dirtiness function 1 above satisfies neither Axiom S2 nor S2′. To see why, consider the databases shown in Fig. 2. Here, $|clusters(DB, \{fd\})| = |clusters(DB', \{fd\})|$, whereas Axiom S2 states that DB' should have a lower dirtiness degree. Now consider DB' without tuples (Mary, 30, 170.5) and (Paul, 35, 171.5). We still have $|clusters(DB, \{fd\})| = |clusters(DB', \{fd\})|$, whereas Axiom S2′ states that DB' should have a lower dirtiness degree.

Proposition 5.3 *Dirtiness function 2 above satisfies Axioms S1 and S2′.*

Unfortunately, dirtiness function 2 above does not satisfy Axiom S2. To see why, consider the databases shown in Fig. 2. In this case, $\sum_{cl \in clusters(DB,\{fd\})} |cl| = \sum_{cl \in clusters(DB',\{fd\})} |cl|$, whereas Axiom S2 states that DB should have a higher dirtiness degree.

5.3 Functions Proposed in the Literature

In this section, we see how certain dirtiness functions proposed in the literature measure up w.r.t. the axioms we have proposed. The following function was proposed in [HK05]:

$$\frac{|culprits(DB, \mathcal{F})|}{|DB \cup \mathcal{F}|}$$

This function looks at the ratio of the total number of culprits to the size of the database and functional dependencies.

Proposition 5.4 *The dirtiness function above satisfies Axiom S1.*

However, this dirtiness function does not satisfy either Axiom S2 nor S2′. The main reason is that this function does not look at the tuples inside a cluster. We consider the two axioms in turn:

(S2) Consider the databases shown in Fig. 3. We have $\frac{|culprits(DB,\{fd\})|}{|DB|+1} = \frac{3}{10}$ and $\frac{|culprits(DB',\{fd\})|}{|DB'|+1} = \frac{1}{3}$, thus contradicting Axiom S2 which states that DB' should have lower dirtiness than DB.

(S2′) The same example used for $S2$ shows that this function does not satisfy Axiom S2′.

DB

Name	*Age*	*Height*
Mary	30	170
Mary	30	170
Matthew	25	153
Matthew	25	153
John	33	163
John	33	163
Paul	36	166
Paul	36	171
Paul	36	174

DB'

Name	*Age*	*Height*
Paul	36	171
Paul	36	174

Figure 3: A case where the Grant-Hunter measure does not satisfy neither axiom Axiom S2 nor S2′

The following dirtiness function was proposed in [GH06]:

$$\frac{|culprits(DB, \mathcal{F})|}{|DB| + |ground(\mathcal{F})|}$$

This function looks at the ratio of the number of culprits to the sum of the size of the database and the number of ground instances of the functional dependencies.

Proposition 5.5 *The dirtiness function above satisfies Axiom S1.*

However, this dirtiness function does not satisfy either Axiom S2 nor S2′ because of the fact that it does not examine clusters. We consider the two axioms in turn:

(S2) Consider the databases shown in Fig. 3. Suppose DB contains $x > 2k-1$ more tuples which do not add to the number of inconsistencies that were already present. In this case we have $\frac{|culprits(DB',\{fd\})|}{|DB'|+k} = \frac{1}{3+k} > \frac{|culprits(DB,\{fd\})|}{|DB|+k}$, thus contradicting Axiom S2 which states that DB' should have lower dirtiness than DB.

(S2′) The same case considered for Axiom S2 shows that Axiom S2' is also contradicted.

The following function was proposed in [HK05, Loz94]:

$$|DB| + |ground(\mathcal{F})| - log_2|\bigcup_{\Delta \in MCS(DB \cup \mathcal{F})} mod(\Delta)|$$

where $MCS(DB \cup \mathcal{F})$ are the maximally consistent subsets of $DB \cup \mathcal{F}$ and $mod(\Delta)$ is the set of models of Δ. This function measures *cleanliness* with respect to functional dependencies, thus Axiom S1 is not applicable. If we take the negative of this function, the resulting dirtiness function does not satisfy Axioms S2 nor S2′. This can easily be seen by observing that adding or removing consistent tuples has a linear impact on the dirtiness measure while not changing the set of clusters.

5.4 A New Single-Dependency Dirtiness Function

Coming up with a single-dependency dirtiness function satisfying the axioms is a challenge. We now propose a new single-dependency dirtiness function δ_{var}. Let DB be a database, fd a functional dependency over DB, $\{A_{fd,1}, \ldots, A_{fd,m}\}$ the sequence of attributes in fd ordered w.r.t. $>_r$, and $variance_{max}(i)$, with $i \in [1, m]$, be the maximum possible variance for attribute $A_{fd,i}$. Let $B > 1$ be any integer. Then:

$$\delta_{var}(DB, fd, >_r) = \sum_{cl \in clusters(DB,\{fd\})} wtVar(cl, fd, >_r)$$

where

$$wtVar(cl, fd, >_r) = \sum_{i=1}^{m} B^{m-i} \cdot var'_{A_{fd,i}}(cl.A_{fd,i});$$

$$var'_{A_{fd,i}}(cl.A_{fd,i}) = (B-1) \cdot \frac{variance_{A_{fd,i}}(cl.A_{fd,i})}{variance_{max}(i)}.$$

Intuitively, we first compute the variance of each attribute $A_{fd,i}$ in each cluster cl, and normalize it to the range $[0, (B-1)]$ (this value is denoted as $var'_{A_{fd,i}}(cl.A_{fd,i})$). Then, for each cluster cl, we sum up the normalized variances of the attributes in fd, with exponentially decreasing weights (with base B) when going from the most reliable attribute to the less reliable one (this sum is denoted as $wtVar(cl, fd, >_r)$). The value of δ_{var} is finally computed as the sum of the $wtVar$'s of all the clusters. The following result says that δ_{var} satisfies all three axioms.

Theorem 5.1 *Function δ_{var} satisfies Axioms S1, S2, and S2'.*

Proof 5.1 *Axiom (S1) is trivial to show. Axiom (S2') immediately follows from Axiom (S2) whose proof we show below.*

Consider two databases DB, DB' such that $DB' \succ_{fd} DB$, and let $\mathcal{P}$ be the set of pairs (cl, cl') such that $cl \in clusters(DB, fd)$, $cl' \in clusters(DB', fd)$, and $cl = \alpha(cl')$ (note that there cannot exist $cl'_1, cl'_2 \in clusters(DB', fd)$ such that $\alpha(cl'_1) = \alpha(cl'_2)$). Moreover, let $\mathcal{P}_{\leq}$, $\mathcal{P}_{<}$, and $\mathcal{S}$ be the sets defined as follows:

- $\mathcal{P}_{\leq} = \{(cl, cl') \in \mathcal{P} \mid cl' \sqsubseteq^{fd}_{var} cl\}$;
- $\mathcal{P}_{<} = \{(cl, cl') \in \mathcal{P} \mid cl' \sqsubset^{fd}_{var} cl\}$;
- $\mathcal{S} = \{cl \in clusters(DB, \{fd\}) \mid \nexists cl' \in clusters(DB', \{fd\}),\ \alpha(cl') = cl\}$.

Note that $\mathcal{P} = \mathcal{P}_{\leq} \cup \mathcal{P}_{<}$. By definition of $\succ_{fd}$, we have that:

- *for each $(cl, cl') \in \mathcal{P}_{\leq}$, $\exists j \in [1, m]$ s.t. $variance_{A_{fd,j}}(cl'.A_{fd,j}) \leq variance_{A_{fd,j}}(cl.A_{fd,j})$, and $\forall i < j$, $variance_{A_{fd,i}}(cl'.A_{fd,i}) = variance_{A_{fd,i}}(cl.A_{fd,i})$;*
- *for each $(cl, cl') \in \mathcal{P}_{<}$, $\exists j \in [1, m]$ s.t. $variance_{A_{fd,j}}(cl'.A_{fd,j}) < variance_{A_{fd,j}}(cl.A_{fd,j})$, and $\forall i < j$, $variance_{A_{fd,i}}(cl'.A_{fd,i}) = variance_{A_{fd,i}}(cl.A_{fd,i})$.*

Since $\forall i \in [0, m]$ $var'_{A_{fd,i}}$ is proportional to $variance_{A_{fd,i}}$, for each $(cl, cl') \in \mathcal{P}$ we have that:

- *the first $j-1$ terms of $wtVar(cl, fd, >_r)$ and $wtVar(cl', fd, >_r)$ are equal;*
- *the j-th terms are multiplied by B^{m-j};*
- *the terms from the $(j+1)$-th to the m-th are multiplied by factors ranging from B^{m-j-1} to 1.*

Thus, as $var'_{A_{fd,i}}$ always ranges between 0 and $B-1$, we have:

- *if $(cl, cl') \in \mathcal{P}_{\leq}$, then $wtVar(cl', fd, >_r) \leq wtVar(cl, fd, >_r)$;*
- *if $(cl, cl') \in \mathcal{P}_{<}$, then $wtVar(cl', fd, >_r) < wtVar(cl, fd, >_r)$.*

Finally, since by definition of $\succ_{fd}$, either $\mathcal{P}_{<}$ or $\mathcal{S}$ are not empty, we have that

$$\delta_{var}(DB, fd, >_r) - \delta_{var}(DB', fd, >_r) =$$
$$\sum_{(cl,cl') \in \mathcal{P}_{\leq}} [wtVar(cl, fd, >_r) - wtVar(cl', fd, >_r)] +$$
$$\sum_{(cl,cl') \in \mathcal{P}_{<}} [wtVar(cl, fd, >_r) - wtVar(cl', fd, >_r)] +$$
$$\sum_{cl \in \mathcal{S}} wtVar(cl, fd, >_r) > 0.$$

Table 1 summarizes which dirtiness functions satisfy which axioms. Note that the only dirtiness function that satisfies all axioms is δ_{var}.

	S1	S2	S2′
$\lvert culprits(DB, \{fd\})\rvert$	✓	×	✓
$\sum_{c \in culprits(DB,\{fd\})} \lvert c\rvert$	✓	×	✓
$\lvert clusters(DB, \{fd\})\rvert$	✓	×	×
$\sum_{cl \in clusters(DB,\{fd\})} \lvert cl\rvert$	✓	×	✓
$\frac{\lvert culprits(DB,\mathcal{F})\rvert}{\lvert DB \cup \mathcal{F}\rvert}$ [HK05]	✓	×	×
$\frac{\lvert culprits(DB,\mathcal{F})\rvert}{\lvert DB\rvert + \lvert ground(\mathcal{F})\rvert}$ [GH06]	✓	×	×
$\lvert DB\rvert + \lvert ground(\mathcal{F})\rvert - log_2 \lvert \bigcup_{\Delta \in MCS(DB \cup \mathcal{F})} mod(\Delta)\rvert$ [HK05, Loz94]	n/a	×	×
δ_{var}	✓	✓	✓

Table 1: Single-dependency dirtiness functions

6 Combining Dirtiness w.r.t. Multiple Functional Dependencies

Most databases will have multiple functional dependencies. In the previous sections, we have looked at the situation where only one functional dependency is present. Combining dirtiness w.r.t. multiple functional dependencies can lead to anomalies.

Example 6.1 *Consider the database in Fig. 4(a) and the following functional dependencies:*

(fd_1) $\forall t, t' \in DB, t.Name = t'.Name \Rightarrow t.Age = t'.Age \wedge t.Salary = t'.Salary \wedge t.Position = t'.Position$

(fd_2) $\forall t, t' \in DB, t.Salary = t'.Salary \Rightarrow t.Position = t'.Position$

Here, $clusters(DB, \{fd_1\}) = \{cl_1, cl_2\}$, *and* $clusters(DB, \{fd_2\}) = \{cl_3\}$. *If we look at the clusters with respect to both functional dependencies, i.e. if we consider* $clusters(DB, \{fd_1, fd_2\})$, *then we obtain the set of all five tuples.*

The following definition specifies what it means for a database to be "clearly cleaner" than another database.

Definition 6.1 *Given a single-dependency dirtiness function* δ, *we say that* $DB' \gtrsim_{\mathcal{F}} DB$, *read "*$DB'$ *is clearly cleaner than* DB *with respect to the set of dependencies* $\mathcal{F}$*", iff* $\forall fd \in \mathcal{F}$, $\delta(DB', fd, >_r) \leq \delta(DB, fd, >_r)$.

Suppose τ is a function that measures the dirtiness of a database DB based on a reliability ordering $>_r$ and a set of functional dependencies, and suppose τ uses a single-dependency dirtiness function δ to measure dirtiness in a database w.r.t. a single functional dependency. Then we hypothesize that τ needs to satisfy the following axiom.

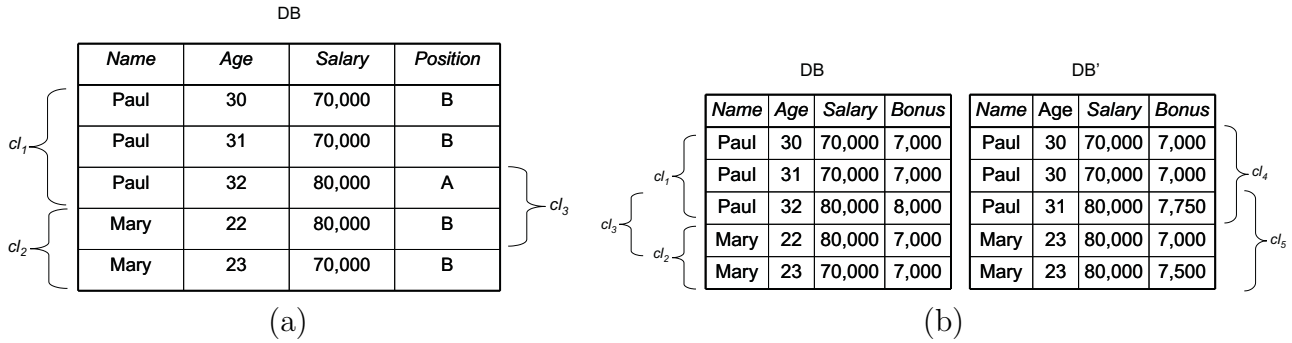

Figure 4: (a) A case where a single cluster may comprise tuples violating different functional dependencies; (b) According to Axiom M1, DB' is clearly cleaner than DB

Axiom M1. If $DB' \gtrsim_{\mathcal{F}} DB$, then $\tau(DB', \mathcal{F}, >_r) \leq \tau(DB, \mathcal{F}, >_r)$.

This axiom merely says that if DB' is clearly cleaner than DB, then τ must assign a lower (or equal) level of dirtiness to DB'.

Example 6.2 *Consider the databases in Fig. 4(b) and the following functional dependencies:*

(fd_1) $\forall t, t' \in DB, t.Name = t'.Name \Rightarrow t.Age = t'.Age \wedge t.Salary = t'.Salary$

(fd_2) $\forall t, t' \in DB, t.Salary = t'.Salary \Rightarrow t.Bonus = t'.Bonus$

Here, $clusters(DB, \{fd_1\}) = \{cl_1, cl_2\}$, $clusters(DB, \{fd_2\}) = \{cl_3\}$, $clusters(DB', \{fd_1\}) = \{cl_4\}$, *and* $clusters(DB', \{fd_2\}) = \{cl_5\}$. *We can clearly see that* $\delta(DB', fd_1, >_r) < \delta(DB, fd_1, >_r)$ *and* $\delta(DB', fd_2, >_r) < \delta(DB, fd_2, >_r)$. *Therefore, Axiom M1 dictates that* $\tau(DB', \mathcal{F}, >_r) \leq \tau(DB, \mathcal{F}, >_r)$.

We now propose two dirtiness functions that support multiple functional dependencies, both of which build on top of a single-dependency dirtiness function. Thus, even though our axioms on multiple-dependency dirtiness functions are weak (because there is only one axiom), things are actually more constrained than might be immediately apparent because they are required to build on top of a single-dependency dirtiness function.

The first function we propose makes the conservative choice of taking the maximum among the values returned by the single-dependency function.

Definition 6.2 (Pessimistic multiple-dependency dirtiness function) *Let DB be a database, $\mathcal{F}$ a set of functional dependencies over DB, $>_r$ an ordering of the attributes of DB, and δ a single-dependency dirtiness function: We define function τ_{max} as*

$$\tau_{max}(DB, \mathcal{F}, >_r) = max_{fd \in \mathcal{F}} \delta(DB, fd, >_r)$$

It is immediate to see that this multiple-dependency dirtiness function satisfies Axiom M1.

Proposition 6.1 *Function τ_{max} satisfies Axiom M1.*

The second dirtiness function takes into account the fact that some functional dependencies might be more important than others, so violations of less important dependencies should contribute less to dirtiness.

Definition 6.3 (Preference-based multiple-dependency dirtiness function) *Let DB be a database, $\mathcal{F}$ a set of functional dependencies over DB, $>_r$ an ordering of the attributes of DB, δ a single-dependency dirtiness function, and $weight : \mathcal{F} \rightarrow \mathbb{N}^+$: We define function τ_{wt} as*

$$\tau_{wt}(DB, \mathcal{F}, >_r) = \frac{\sum_{fd \in \mathcal{F}} weight(fd) \cdot \delta(DB, fd, >_r)}{\sum_{fd \in \mathcal{F}} weight(fd)}$$

The following straightforward result says that τ_{wt} also satisfies Axiom M1.

Proposition 6.2 *Function τ_{wt} satisfies Axiom M1.*

A special case of τ_{wt} takes the average of the dirtiness values returned by the single-dependency function:

$$\tau_{avg}(DB, \mathcal{F}, >_r) \quad = \quad \frac{\sum_{fd \in \mathcal{F}} \delta(DB, fd, >_r)}{|\mathcal{F}|}$$

obtained by setting in τ_{wt}, $\forall fd \in \mathcal{F}$, $weight(fd) = k$ for any fixed $k \in \mathbb{N}^+$.

7 Related Work and Conclusions

There has been a tremendous amount of work in inconsistency management since the 60s and 70s when paraconsistent logics where introduced [dC74] and logics of inconsistency [Bel77, Gra78] were developed. Subsequently, frameworks such as default logic [Rei80], maximal consistent subsets [BKMS92] and inheritance networks [Tou86] and others were used to generate multiple plausible consistent scenarios (often called "extensions"), and methods to draw inferences were developed that looked at truth in all (or some) extensions. Argumentation methods [AC02] were used to reason about how certain arguments defeated others. Methods to clean data and/or provide consistent query answers in the presence of inconsistent data are also quite common [JDR99, SS03, Cho07, BFFR05]. For instance, [Cho07] addresses the basic concepts and results of the area of consistent query answering (in the standard model-theoretic sense). They consider universal and binary integrity constraints, denial constraints, functional dependencies, and referential integrity constraints. [BFFR05] presents a cost-based framework that allows finding "good" repairs for databases that exhibit inconsistencies in the form of violations to either functional or inclusion dependencies. They propose heuristic approaches to constructing repairs based on equivalence classes of attribute values; the algorithms presented are based on greedy selection of least repair cost, and a number of performance optimizations are also explored.

However, we are aware of very few works on measuring the degree of inconsistency in a database. All three methods deal with culprits only or with maximal consistent subsets [Loz94, HK05, GH06]. We believe we have made two important conceptual contributions in this paper. First, we draw attention to the notion of a *cluster* and explain that clusters are very important in measuring cleanliness of the database. Second, we have drawn attention to the fact that well known statistical measures for measuring variation in a dataset (such as standard deviation and variance) have a role to play in measuring the dirtiness of a database. Based on these two ideas, we have developed single-dependency axioms that we believe a dirtiness measure should satisfy when one functional dependency is considered in isolation. We subsequently look at some obvious dirtiness measures based on culprits and clusters, as well as past work, and show that these methods do not satisfy our axioms. We then develop our own dirtiness measure that satisfies these axioms. Subsequently, we propose a single axiom for dirtiness functions that handle multiple functional dependencies – however, such dirtiness functions are supposed to be built on top of a dirtiness function for single dependencies. We present a couple of alternative dirtiness functions that satisfy this axiom.

Future work will focus on the development of other multiple-dependency dirtiness functions and experimental evaluations of how these dirtiness functions work in practice in terms of computational overhead they impose. Moreover, we plan to build "cleaning" operators that provably reduce dirtiness.

References

[AC02] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1–3):197–215, 2002.

[BDP+02] P. Bosc, D. Dubois, O. Pivert, H. Prade, and M. de Calmes. Fuzzy summarization of data using fuzzy cardinalities. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, pages 1553–1559, 2002.

[Bel77] N. Belnap. A useful four valued logic. *Modern Uses of Many Valued Logic (eds. G. Epstein and M. Dunn)*, pages 8–37, 1977.

[BFFR05] Philip Bohannon, Wenfei Fan, Michael Flaster, and Rajeev Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *SIGMOD*, pages 143–154, 2005.

[BKMS92] C. Baral, S. Kraus, J. Minker, and V.S. Subrahmanian. Combining knowledge bases consisting of first order theories. *Computational Intelligence*, 1992.

[Cho07] Jan Chomicki. Consistent query answering: Five easy pieces. In *ICDT*, pages 1–17, 2007.

[dC74] N.C.A. da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15(4):497–510, 1974.

[GH06] John Grant and Anthony Hunter. Measuring inconsistency in knowledgebases. *J. Intell. Inf. Syst.*, 27(2):159–184, 2006.

[Gra78] John Grant. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic*, 19(3):435–444, 1978.

[HK05] Anthony Hunter and Sébastien Konieczny. Approaches to measuring inconsistent information. In *Inconsistency Tolerance*, pages 191–236, 2005.

[JDR99] Paul Jermyn, Maurice Dixon, and Brian J. Read. Preparing clean views of data for data mining. In *ERCIM Workshop on Database Research*, pages 1–15, 1999.

[Loz94] E. L. Lozinskii. Resolving contradictions: A plausible semantics for inconsistent systems. *J. of Automated Reasoning*, 12(1):1–31, 1994.

[MPS+07] Maria Vanina Martinez, Andrea Pugliese, Gerardo I. Simari, V. S. Subrahmanian, and Henri Prade. How dirty is your relational database? an axiomatic approach. In *ECSQARU*, pages 103–114, 2007.

[Rei80] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 1980.

[SS03] E. Schallehn and K. Sattler. Using Similarity-based Operations for Resolving Data-level Conflicts. In A. James, B. Lings, and M. Younas, editors, *British National Conference on Databases*, volume 2712 of *lncs*, pages 172–189, Berlin, 2003. Springer-Verlag.

[Tou86] D. Touretzky. *The mathematics of inheritance systems*. Morgan Kaufmann, 1986.

[Ull89] Jeff Ullman. *Principles of Data Base and Knowledge Base Systems*. Addison Wesley, 1989.