Threshold Functional Dependencies for Time Series Data

Mingyue Ji[1], Xiukun Wei[2], and Dongjing Miao[1](✉)

[1] Harbin Institute of Technology, Harbin 150001, Heilongjiang, China
jimy2116@mails.jlu.edu.cn, miaodongjing@hit.edu.cn
[2] Jilin University of Science and Technology, Changchun 130012, Jilin, China
weixk2116@mails.jlu.edu.cn

Abstract. This paper extends traditional Functional Dependencies (FDs) to Threshold Functional Dependencies (TFDs) for Time Series Database according to the characteristics of attribute values changing rapidly by time from sensors. In contrast to the *unique-to-same* pattern in relational schema, TFDs allow determined attribute value within a certain range rather than a clear value when corresponding to the same deciding party. We find that TFDs capable of not only detecting errors resulting from attribute value out-of-bounds in one tuple horizontally, but also from a column of single attribute among several tuples vertically. And we focus more on the former in this article. We draw a clear line between FDs and TFDs because they have some intersection. And we classify TFDs for convenience of research. We provide an inference system for classified TFDs analogous to Armstrong's axioms, prove its soundness and completeness and explain their differences and connections. We perform some experiments to show effects of TFDs which make some contributions to data quality for Time Series Database.

Keywords: Time series · Functional dependency · Threshold

1 Introduction

Time Series Database (TSDB) is mainly applied to industrial monitoring such as electric power, petroleum, chemical industry, etc. It is adept at processing constantly updated and rapidly changing data and transaction with time limits. There are inevitable errors in these sensor readings. Integrity constraints in TSDB differ from relational database mainly on functional dependencies. If we say attribute values in relational database are mostly enum types, then attribute values in TSDB are mostly continuous data in their domain but are recorded discretely. And functional dependency like A → B in the former is equality because every B value determined by the unique A must equal, while the latter focus on inequality where B can change within a reasonable range in the same condition. These differences are enough to give us a motivation to create a new setup FDs for TSDB.

Variants of FDs have arisen on a small but useful and promising scale which provide more strict or lighter constraints among attributes in the previous research. Wenfei Fan

Y. Nah et al. (Eds.): DASFAA 2020 Workshops, LNCS 12115, pp. 164–174, 2020.
https://doi.org/10.1007/978-3-030-59413-8_14

[1] et al. propose conditional functional dependencies (CFDs) aiming at capturing the consistency of data with satisfaction of certain attribute's value as a constant. The formula is $(X \rightarrow Y, T_p)$ where T_p is the pattern keeping certain attribute constant. Flip Korn [2] et al. define FD probabilistic, approximate constraints (FDPACs) for network traffic database by converting $X \rightarrow Y$ to $f(x) \rightarrow g(x)$. It achieves by allowing an aggregate f over a set of attribute values in X to functionally determine the similar aggregate g over those in Y. R. Haux and U. Eckert [3] introduce non-deterministic functional dependencies (NFDs) for inherent connections among attributes that can be emphasized by time. For example, one patient's weight varies with hormone level at several examination dates denoting that the patient's ID determines his non-deterministic and random weight variable. It can be expressed as $ID \rightarrow \Phi(Weight)$. So NFDs can be regarded as stochastic extensions of traditional FDs. NFDs make some difference among the above state-of-art because of correlation with time, which relates to TSDB.

Inspired by the above variants of FDs, we bring the concept of our special FDs for TSDB. When applying FD $X \rightarrow Y$ to TSDB, we find that each attribute value in Y can come from a reasonable range as t_1, t_2 and t_3 in Table 1 show with A determining B's range. We can see that t_2 is correct while t_1 and t_3 violate the range. These ranges can depend on single attribute or combined attributes restricted by nature or machinery (e.g., outdoor temperature in different places), statistic analysis methods (e.g. linear or polynomial regression), sampling frequency and so on. The range maybe includes random (discrete) variables, functions or even a constant with zero-range. Regardless of types of this range, we focus more on logic on the data schema level rather than the upper statistics methods. We name this kind of FDs Threshold Functional Dependencies (TFDs). Threshold literally seems to resemble domain integrity but they are different. We will discuss it later. As shown by the Arrow 1 hinting in Table 1, we say TFDs have function of horizontal constraints embodying in rows in Table 1.

Table 1. An instance for threshold FD

	Time	A	Arrow 1	B
t_1	10:01	5.1		$4.8 \notin [5.1-1,5.1+1]$
t_2	10:02	5.2		$5.3 \in [5.2-1,5.2+1]$
t_3	10:03	5.6		$11.0 \notin [5.6-1,5.6+1]$
t_4	10:04	50.4		$49.8 \in [50.4-1,50.4+1]$
t_5	10:05	6.6	Arrow 2	$5.9 \in [6.6-1,6.6+1]$

From other perspectives to see errors in TSDB, outlier (or anomaly) detection [4] and repairing have been studied well in TSDB. The main ideas are based on techniques such as AR model [5], Markov models [6], neural network [7] and so on. Clearly, these statistical techniques reflect the invisible constraints by time on single attribute values in a column in a table. As shown by the Arrow 2 hinting in Table 1, we name it vertical constraint. Obviously, time stamp itself is a typical vertical constraint and Shaoxu Song

[8] et al. have studied the time stamp repairing problem. These statistical techniques above can detect t_4 which is an abrupt change among t_1 to t_5 in Table 1.

Actually, not only can our TFDs express horizontal but also vertical constraints when we use statistical techniques above to restrict different values or their ranges of a column of single attribute changing with time. In this article, we focus more on TFD's horizontal constraints rather than its vertical constraints.

Proposing a new variant of traditional functional dependencies named TFDs according with detecting errors in time series database is our first contribution. The capacity of detecting TFDs covers horizontal and vertical constraints, which is our second contribution. The third contribution is classifying TFDs and presenting a sound and complete inference system for classified TFDs analogous to Armstrong's axioms. The last contribution is that the experiments show the performance of TFDs when detecting errors in time series database.

2 Threshold Functional Dependencies

Here, we will define TFDs. Consider a Time Series Database T and schema R over a set of attributes, denoted by attr(R). And we denote attributes in R as A, B, $C\ldots$ and sets of attributes of X, Y, $Z\ldots$

2.1 Definition

Definition 1. A Threshold FD over R is the form $X \rightarrow \Gamma(Y)$ where Γ represents the threshold of every attribute's values in Y.

We refer to X as the left-hand side, or *lhs* for short, and to Y as the right-hand side, or *rhs* for short. If one tuple t satisfies a TFD Δ in horizontal constraints, that means attribute values in Y are within the Γ range, denoted by $t \vdash \Delta$. If every tuple in table T satisfies each element in TFD set Λ in horizontal constraints, we say $T \vdash \Lambda$.

Threshold Γ represents value ranges for *rhs* decided by *lhs* with upper and lower bounds in one or several intervals. We denote these bounds as *threshold functions*. Though we have mentioned we don't want to care more about the format of *threshold functions*, we need examples to illustrate TFDs as follows. **e.g.1**, $A, B \rightarrow \Gamma(C, D)$, $\Gamma = \{C \in (f_1(A), f_2(A)], D \in [f_3(A, B), f_4(A, B)), D \in [f_5(A, B), f_6(A, B))\}$, and f_1 to f_6 are *threshold functions*. They can be implicit or explicit functions for some attribute.

Assuming A_e is an expression only containing an arbitrary attribute A from $X \cup Y$, if A_e can be expressed by $(X \cup Y)\backslash A$ explicitly, we denote A as *A-ex*. **e.g.2**, if $A \rightarrow \Gamma(B)$, $\Gamma = \{B \in [-A - A^2, A + A^2]\ (A \geq 0$ and B is integer$)\}$, we have *B-ex* and *A-ex*. If A_e is trapped in the implicit *threshold functions*, we denote A as *A-im*. **e.g.3**, if $A, B \rightarrow \Gamma(A, B)$, $\Gamma = \{g_1(A, B) \leq 10\}$, $g_1 = A^3 - \sqrt{A} + A^2B^3 - A\sqrt{B}$, we have *A-im* and *B-im*. Specially we call the attribute set *AB* in **e.g.3** as *AB-ex* with *A-im* and *B-im*, and call attribute set *CD* in **e.g.1** as *CD-ex* with *C-ex* and *D-ex*. We can see if an attribute set C is *C-ex*, then arbitrary element A of C can be *A-ex* or *A-im*. Formally speaking, an attribute set C is *C-ex* if and only if C can be expressed by attributes except any element of C or can be expressed only by C itself as **e.g.3** shows. So when a proper subset S of C is *S-ex*, then there must exist *C\S-ex* according to symmetry.

It is not necessary that everyone of *lhs* must participate in decision on *rhs*, as C's range in **e.g.1** shows, in which case we denote B as C-*irre*. If some attribute from *lhs* is irrelevant to everyone of *rhs*, we say the attribute is Y-*irre*. In $X \rightarrow \Gamma(Y)$, we don't allow X are all Y-*irre*. In other words, there at least exists one attribute which is not Y-*irre*. Sometimes there is exactly no mathematical formula between *lhs* and *rhs* due to multi-implicit, but they do have some connections. We denote this situation as *vague threshold*, and this TFD as false-TFD. We will give an example of *vague threshold* later. If X determine Y by *vague threshold*, we don't consider them into the following research in this article.

2.2 Classifying

We classify TFD $X \rightarrow \Gamma(Y)$ into two cases according to the kinds of Γ.

Case 1 Γ does not Function
$X \rightarrow \Gamma(Y)$ naturally becomes $X \rightarrow Y$, which is exactly a traditional FD, as hinted by Fig. 1(a), which is called *unique-to-same*. And the threshold is zero, denoting as *empty-threshold*. Data consistency guarantees that tuples with every attribute in common value in X must have the same value set of Y where each attribute value in Y has a countably infinite domain. **e.g.4,** $A \rightarrow \Gamma(A)$, $\Gamma = \{A \in [A - 0, A + 0]\}$. We classify the case of **e.g.4** into TFDs.

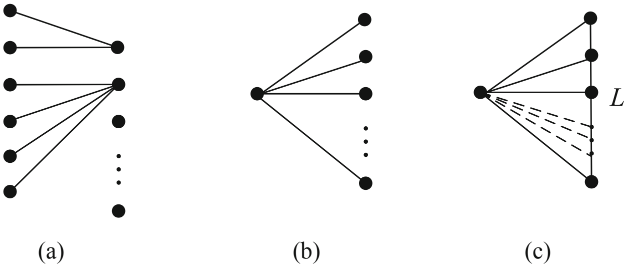


Fig. 1. Sketch showing patterns of FDs and TFDs

We hereby declare though from the perspective of concept and definition, TFDs contains part of traditional FDs, we won't mix them into one Γ in this article for the convenience of exploiting TFDs. So now our research scope is TFD including the case of **e.g.4**.

Case 2 Γ Maps More Values for Everyone in Y in the Same *lhs* Situation
The dependency becomes more slack relative to case 1. Tuples with every attribute in common value in X which have different value sets of Y remain existing when performing data cleaning. The value domain of every attribute in Y may be countably infinite or finite, as hinted by Fig. 1(b), and as B's threshold in **e.g.2** shows. It may be non-countably continuous, as hinted by Fig. 1(c) where the nodes can slide along the straight line L if we allow B not must be integrity in **e.g.2**. Specially, when $A = 0$ in **e.g.2**, though B can only be zero, it is different from case 1 because of dynamic variability of A.

2.3 Properties of TFDs

Here, we describe the special characteristics of TFDs. At the same time we explain the differences and connections between TFDs and traditional FDs, also between TFDs and domain constraints.

- The consistency restricted by traditional FDs must be reflected in more than two tuples. But one single tuple can accord with or violate a TFD in horizontal constraints.
- Every traditional FD set can have its corresponding consistent table before or after repairing. So data schema over traditional FDs is always worth repairing but it is not the case over TFDs. Not every TFDs set Λ can make sense due to the range of *rhs*. **e.g.5**, $A \rightarrow \Gamma_1(C)$, $\Gamma_1 = \{C \in [1/A, +\infty]\ (A > 0)\}$ and $B \rightarrow \Gamma_2(C)$, $\Gamma_2 = \{C \in [-\infty, -1/B](B > 0)\}$. We can see there doesn't exist one correct C value at all.
- As mentioned earlier, TFDs appear literally to be like domain constraints but they are totally different. The former imposes restrictions to certain column data determined by other attributes while the latter limits every column data and has nothing to do with other attributes.

3 An Inference System for TFDs

3.1 TFD Rules

There is an inference system for TFD analogous to Armstrong's Axioms.

Lemma 1. $A \rightarrow \Gamma(A)$

This TFD always holds with *empty-threshold* $\Gamma = \{A \in [A - 0, A + 0]\}$. $A \rightarrow \Gamma(A)$ makes sense in both traditional FDs and TFDs. In case 1, $A \rightarrow \Gamma(A)$ has been classified into TFD. Specially, assuming $X = \{A, B\}$, $X \rightarrow \Gamma(X)$ holds not necessarily due to $A \rightarrow \Gamma(A)$ and $B \rightarrow \Gamma(B)$. $X \rightarrow \Gamma(X)$ holds may due to $\Gamma = \{A \in [-B, B], B \in [-A, A]\}$. Moreover, **e.g.3** also belongs to this case with *AB-ex. So* we can conclude that $A \rightarrow \Gamma(A)$ is not necessity and substitute for the validity of $X \rightarrow \Gamma(X)$.

TFD1. If $Y \subseteq X \subseteq R$, then $X \rightarrow \Gamma(Y)$.
TFD1 corresponds to Armstrong's reflexivity.
TFD2. If $X \rightarrow \Gamma_1(Y)$ and $Z \subseteq R$, then $XZ \rightarrow \Gamma_1(Y)$ and $XZ \rightarrow \Gamma_2(YZ)$.
TFD2 extends Armstrong's augmentation. This rule emphasizes not every one of *lhs* must determine *rhs*. Actually these redundant items are encouraged to be cleaned.
TFD3. If $X \rightarrow \Gamma_1(Y)$ and $Z \rightarrow \Gamma_2(W)$, then $XZ \rightarrow \Gamma_3(YW)$.
TFD3 also extends Armstrong's augmentation.
TFD4. If $X \rightarrow \Gamma_1(Y)$, and $Z \subseteq Y$ with *Z-ex*, then $X \rightarrow \Gamma_2(Z)$.
TFD4 is similar to the decomposition corollary in Armstrong's axioms.
TFD5. If $XY \rightarrow \Gamma(Z)$, X and Y affect Z respectively, then $X \rightarrow \Gamma_1(Z)$ and $Y \rightarrow \Gamma_2(Z)$.
TFD6. If $XY \rightarrow \Gamma(Z)$ with $W \subseteq XY$, and W doesn't work on Z range, then $XY \backslash W \rightarrow \Gamma(Z)$.

For TFD5 and TFD6, there are no counterpart axioms for Armstrong's traditional FDs. And they are complementary to TFD2. If we apply TFD5 and TFD6 to a TFD, **e.g.1**, $A, B \rightarrow \Gamma(C, D)$, $\Gamma = \{C \in (f_1(A), f_2(A)], D \in [f_3(A, B), f_4(A, B)), D \in [f_5(A, B), f_6(A, B))\}$, then we can obtain by TFD5 $A, B \rightarrow \Gamma_1(C)$, $\Gamma_1 = \{C \in (f_1(A), f_2(A)]\}$ and $A, B \rightarrow \Gamma_2(D)$, $\Gamma_2 = \{D \in [f_3(A, B), f_4(A, B)), D \in [f_5(A, B), f_6(A, B))\}$. And by TFD6, the former one becomes $A \rightarrow \Gamma_3(C)$, $\Gamma_3 = \{C \in (f_1(A), f_2(A)]\}$.

Specially, TFD2, TFD5 and TFD6 are contrary to Armstrong's merger and decomposition corollaries, servicing for *lhs* and *rhs* respectively.

TFD7. If $X \rightarrow \Gamma(Y)$ with $Ram \subseteq X \cup Y$, and Ram is *Ram-ex*, then $X \cup Y \rightarrow \Gamma_i(Ram)$.

There is not an issue for traditional FDs corresponding to TFD7. Actually TFDs mean inequalities, so some attributes can become independent variables in *threshold functions* with others becoming dependent variables. The dependent variables are exactly the *rhs*. It seems like pulling one hair and moving the whole body.

In TFDs, we don't have Armstrong's transitivity counterpart. Without loss of generality, transitivity in TFDs will bring *vague threshold*. The illustration is as follows. If one tuple t satisfies $X \rightarrow \Gamma_1(Y)$ and $Y \rightarrow \Gamma_2(Z)$, then we know that $t[Z]$ is within the Γ_2 range decided by $t[Y]$ which is decided by $t[X]$. So in the same tuple, $t[X]$ affects $t[Z]$'s value.

But there doesn't always exist a non-false-TFD between X and Z due to implicit bridge Y. **e.g.6**, $A \rightarrow \Gamma_1(B)$, $\Gamma_1 = \{B \in [A - 10, A\ \ 10]\}$, $B \rightarrow \Gamma_2(C)$, $\Gamma_2 = \{C \in [\mathrm{B}, + \infty]\}$, so we can deduce $A \rightarrow \Gamma_3(C)$, $\Gamma_3 = \{C \in [A + 10, + \infty]\}$. And **e.g.7**, if we change the first TFD above into $A \rightarrow \Gamma_1(B)$, $\Gamma_1 = \{B * \ln B + 1/B \in [A - 10, A + 10]\}$, then the last TFD above comes to $A \rightarrow \Gamma_3(C)$ with *vague threshold* Γ_3, and this is the case of false-TFD. Sometimes we attain a TFD by transitivity as **e.g.6** shows, but the result is false-TFD more often as **e.g.7** shows. For convenience and unity, we don't accept transitivity of TFDs.

3.2 Sound and Complete

Here we prove soundness and completeness of the inference system.

Definition 2. Let Λ be a set of TFDs. If $X \rightarrow \Gamma(Y)$ is deduced from Λ using TFD rules above, then we say $\Lambda \vDash X \rightarrow \Gamma(Y)$.

Definition 3. Let Φ be a set of inference rules {TFD1 to TFD7}. Then Φ is sound for logical implication of TFDs if $X \rightarrow \Gamma(Y)$ is deduced from Λ using Φ, and $X \rightarrow \Gamma(Y)$ is true in any relation in which the TFDs of Λ are true.

Lemma 2. Φ is sound for logical implication of TFDs.

Proof. In order to prove the soundness of Φ we have to prove that each of the TFD rules is sound. The processes are Proof 1 to Proof 7.

Proof 1. If $Y \subseteq X \subseteq R$, then $X \rightarrow \Gamma(Y)$.

For each attribute in Y, denoted as Ram, $Ram \rightarrow \Gamma(Ram)$ holds according to Lemma 1. $Ram \rightarrow \Gamma(Ram)$ guarantees that when $X \backslash Ram$ don't determine *Ram, there* at least exists one *threshold function* for *Ram* to hold for Γ. **e.g.8**, $AB \rightarrow \Gamma(AB)$, $\Gamma = \{A \in [A - 0, A + 0], B \in [B - 0, B + 0]\}$. When *X\Ram* decide *Ram's* values by some *threshold functions*, $Ram \rightarrow \Gamma(Ram)$ is right but doesn't work in Γ as we discuss in Lemma 1. **e.g.9**, $AB \rightarrow \Gamma(AB)$, $\Gamma = \{A \in [B - 1, B + 1], B \in [A - 1, A + 1]\}$.

Proof 2. If $X \rightarrow \Gamma_1(Y)$ and $Z \subseteq R$, then $XZ \rightarrow \Gamma_1(Y)$ and $XZ \rightarrow \Gamma_2(YZ)$.

If one tuple t satisfies $X \rightarrow \Gamma(Y)$, then $t[Y]$ is within the Γ range decided by $t[X]$, also by $t[XZ]$ considering Z as *Y-irre*. So $XZ \rightarrow \Gamma(Y)$ holds with *threshold functions* unchanged. In the same tuple, t[Z]'s value range also can be determined by t[Z] exactly as Lemma 1 shows, considering X as *Z-irre*. So $XZ \rightarrow \Gamma(YZ)$ holds with added *threshold functions* like in **e.g.8**.

Proof 3. If $X \rightarrow \Gamma_1(Y)$ and $Z \rightarrow \Gamma_2(W)$, then $XZ \rightarrow \Gamma_3(YW)$.

If one tuple t satisfies $X \rightarrow \Gamma_1(Y)$ and $Z \rightarrow \Gamma_2(W)$, then $t[Y]$ is within the Γ_1 range decided by $t[X]$ and $t[W]$ is within the Γ_2 range decided by t[Z]. In the same tuple, we can say $t[Y]$ and $t[W]$ are within respective ranges decided by $t[X]$ and t[Z] combined in Γ_3. So $XZ \rightarrow \Gamma_3(YW)$ holds.

Proof 4. If $X \rightarrow \Gamma_1(Y)$, and $Z \subseteq Y$ with *Z-ex*, then $X \rightarrow \Gamma_2(Z)$.

If one tuple t satisfies $X \rightarrow \Gamma_1(Y)$ and $Z \subseteq Y$ with *Z-ex*, we can convert it into $X \rightarrow \Gamma_1(Z \cup (Y\backslash Z))$. In tuple t, we can say t[Z] and t[Y\Z] are both within the Γ_1 range decided by $t[X]$. We divide *threshold functions* in Γ_1 into Γ_2 and Γ_3 for Z and Y\Z respectively. So we can say in the same tuple t, t[Z] is within the Γ_2 range decided by $t[X]$ while t[Y\Z] is within Γ_3. So $X \rightarrow \Gamma_2(Z)$ and $X \rightarrow \Gamma_3(Y\backslash Z)$ hold.

Proof 5. If $XY \rightarrow \Gamma(Z)$, X and Y affect Z respectively, then $X \rightarrow \Gamma_1(Z)$ and $Y \rightarrow \Gamma_2(Z)$.

If one tuple t satisfies $XY \rightarrow \Gamma(Z)$, X and Y affect Z range respectively, then $t[Z]$ is within the Γ range decided by $t[X]$ and by $t[Y]$ respectively. So we can say $X \rightarrow \Gamma_1(Z)$ and $Y \rightarrow \Gamma_2(Z)$ hold. **e.g.10**, if $A, B \rightarrow \Gamma(C)$, $\Gamma = \{C \in [f_1(A), f_2(A)], C \in [f_3(B), f_4(B)]\}$, then $A \rightarrow \Gamma_1(C)$, $\Gamma_1 = \{C \in [f_1(A), f_2(A)]\}$ and $B \rightarrow \Gamma_2(C)$, $\Gamma_2 = \{C \in [f_3(B), f_4(B)]\}$ hold.

Proof 6 If $XY \rightarrow \Gamma(Z)$ with $W \subseteq XY$, and W doesn't work on Z range, then $XY\backslash W \rightarrow \Gamma(Z)$.

If one tuple t satisfies $XY \rightarrow \Gamma(Z)$ with $W \subseteq XY$, and W doesn't work on Z range, then t[Z] is within the Γ range decided by $t[XY\backslash W]$. So $XY\backslash W \rightarrow \Gamma(Z)$ holds. **e.g.11**, if $A, B \rightarrow \Gamma(C)$, $\Gamma = \{C \in [f_1(A), f_2(A)]\}$, then we have $A \rightarrow \Gamma(C)$, $\Gamma = \{C \in [f_1(A), f_2(A)]\}$.

Proof 7. If $X \rightarrow \Gamma(Y)$ with $Ram \subseteq X \cup Y$, and *Ram* is *Ram-ex*, then $X \cup Y \rightarrow \Gamma_i(Ram)$.

If one tuple t satisfies $X \rightarrow \Gamma(Y)$, then $t[Y]$ is within the Γ range decided by $t[X]$. Denoting some elements in $X \cup Y$ as *Ram, if Ram* is *Ram-ex*, then t[*Ram*]'s value is affected by attributes $X \cup Y\backslash Ram$ or *Ram* itself like in Lemma 1. So we have $X \cup Y \rightarrow$

$\Gamma_i(Ram)$ due to permissible redundant of *lhs* implemented by TFD2. **e.g.12**, A, $B \to \Gamma_1(C)$, $\Gamma_1 = \{C \in [A + B - 10, A + B + 10]\}$, so A, $C \to \Gamma_2(B)$, $\Gamma_2 = \{B \in [C - A - 10, C - A + 10]\}$ and B, $C \to \Gamma_3(A)$, $\Gamma_3 = \{A \in [C - B - 10, C - B + 10]\}$. Or we have **e.g.3** as example. But when *Ram* is not *Ram*-ex, **e.g.13**, A, $B \to \Gamma_1(C)$, $\Gamma_1 = \{C \in [B + \mathrm{e}^{AB} + A - 10, B + \mathrm{e}^{AB} + A + 10]\}$, then we cannot get A, $C \to \Gamma_2(B)$ or B, $C \to \Gamma_3(A)$ but A, $B \to \Gamma(C)$ is correct.

Definition 4. Let Φ be a set of inference rules {TFD1 to TFD7}. Let Λ + be all TFDs which can be deduced by Λ using Φ and Λ itself. Φ is complete if every element of Λ + can be deduced by starting from Λ and reasoning from Φ.

It is NP-hard to find out all TFDs in Λ+ as in Armstrong's axioms for traditional FDs. So we prove TFD rules' completeness by contrapositive way.

Proposition 1. If $X \to \Gamma(Y)$ can never be deduced from Λ by using Φ, then $X \to \Gamma(Y)$ can never be implicated in Λ+.

Lemma 3. The inference rules TFD1 to TFD7 are complete.

Proof. Contrapositive of TFDs rules' completeness, that is Proposition 1, proves Lemma 3.

Theorem 1. The inference system is sound and complete.

4 Experiments

In this section, we present results of detecting TFDs horizontal violations over climate data by algorithm shown in Fig. 2.

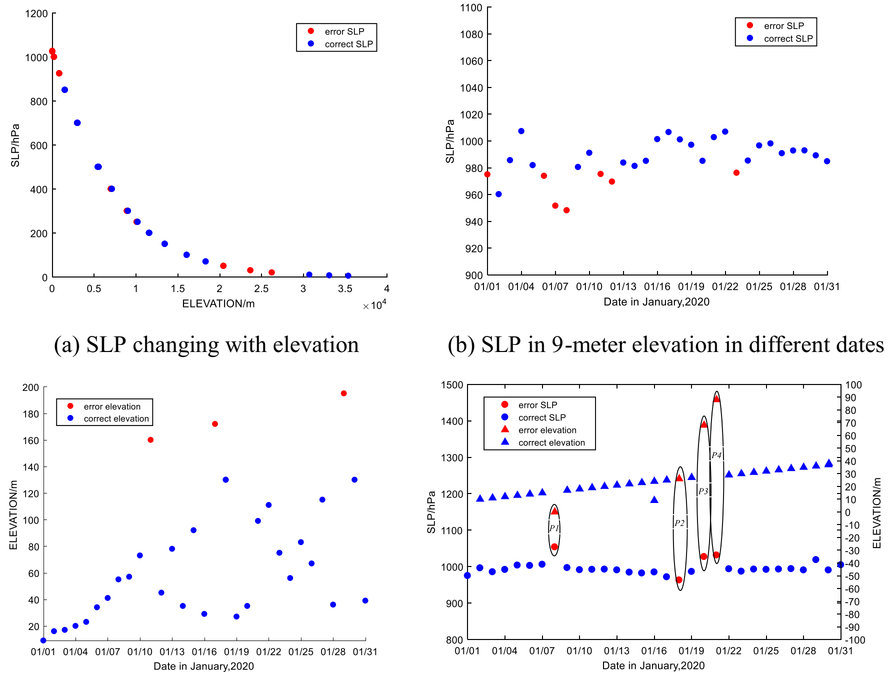
```
Algorithm Detecting_horizon
Input:  a set of TFDs Λ,a TSDB table T
Output: tuples err[n] with error data
for each TFD Δ in Λ do
    Remove irrelevant attributes to rhs in lhs by TFD6.
end for
for each tuple t in T do
    for each TFD Δ in Λ do
        for each formula Γ in Δ do
            attr[R]←attributes in Γ.
            if attr[R] does not satisfy Γ then
                err[n]←t.
            end if
        end for
    end for
end for
```

Fig. 2. Algorithm detecting horizontal errors

We find a TFD between Sea Level Pressure (SLP) and elevation in climate series data based on (1) where p represents SLP/Pa and h represents elevation/m.

$$p = e^{5.25885\times \mathrm{1n}(288.15-0.0065h)-18.2573} \tag{1}$$

So from (1) we obtain a TFD $h \rightarrow \Gamma(p)$, $\Gamma = \{p \in [(1) - \delta, (1) + \delta\}$. Though the above equality is deduced in the condition of considering influence of atmosphere and temperature, there must exist a confidence interval in natural circumstances. In our experiments, we let δ be 35 and intuitive detecting-errors results are as Fig. 3(a), (b), (c) show.



(a) SLP changing with elevation

(b) SLP in 9-meter elevation in different dates

(c) elevation in 996 to 997hPa SLP in different dates

(d) SLP and elevation in different dates

Fig. 3. Experimental results

It is worth mentioning that we execute the algorithm in [9] which can detect abrupt changes with time to obtain errors among tuples while performing *detecting_horizon* algorithm, for which Fig. 3(d) shows horizontal and vertical constraints at the same time. Time node P_1 doesn't satisfy both constraints while P_2 satisfies the latter but not the former and P_3, P_4 are exactly opposite to P_2.

From the above results, we can see that there are several types of errors occurred in TSDB, those satisfy or don't satisfy TFDs with or without abrupt changes. If one attribute A is an abrupt change in a column data, then other attributes in TFD including A are greatly possibly abrupt changes because of value bindings in TFD.In this case, we just need to check vertical constraints only in the column of attribute A and its corresponding TFDs to save time and energy. So TFDs reduce the burden of vertical detection of every column data, as Fig. 4 shows, and TFDs can detect errors between attributes in one tuple that cannot achieved in vertical detection.

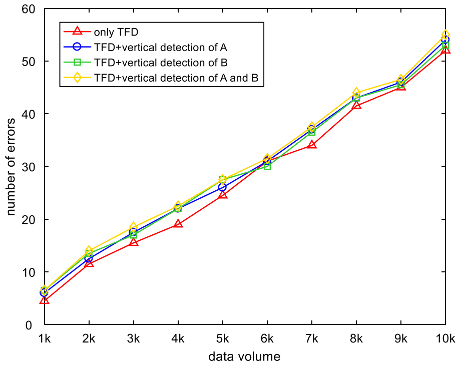


Fig. 4. Detecting errors in different settings of techniques

If we classify types of errors in TSDB with the same probability of each error type by satisfying or not satisfying with or without abrupt changes, we can obtain $2^{n+1}-1$ errors when there are n attributes in one TFD. Under the condition of implementation of the single TFD with n attributes, the rates of error types that can be detected by arbitrary number of attributes' vertical detection are shown as Table 2.

Table 2. The rates of error types that can be detected

Error types	Only detecting TFD	Detecting TFD and vertical constraints in one attribute	Detecting TFD and vertical constraints in two attributes	Detecting TFD and vertical constraints in k attributes ($k \leq n$)
$2^{n+1}-1$	$2^n/2^{n+1}-1 \geq$ 50%	$2^n+2^{n-1}/2^{n+1}-1 \geq 75\%$	$2^n+2^{n-1}+2^{n-2}/2^{n+1}-1 \geq$ 87.5%	$2^n+2^{n-1}+ \ldots + 2^{n-k}/2^{n+1}-1$

Actually, numbers of errors satisfying TFD with abrupt changes such as P_3 an P_4 in Fig. 3(d) are less than other types, so rates in the above table will be arose in practice.

5 Conclusions

We propose a new variant of traditional Functional Dependencies in this article and provide classified TFDs with some rules for its logical operation. But the theoretical system is not that perfect. We have mentioned the most different content contrasting to traditional FD's system is the correctness of TFDs set. So next we will focus on this problem and find effective algorithm to judge correctness of a TFD set. In addition, the vertical constraint hidden in TFDs has been still unsolved though we simply show its capacity to detect errors in the experiments. It will be a key point of our future work.

References

1. Bohannon, P., Fan, W., Geerts, F., Jia, X., Kementsietsidis, A.: Conditional functional dependencies for data cleaning. In: IEEE 23rd International Conference on Data Engineering, ICDE 2007, pp. 746–755 (2007)
2. Korn, F., Muthukrishnan, S., Zhu, Y.: Checks and balances: monitoring data quality problems in network traffic databases. In: Proceedings of 29th International Conference on Very Large Data Bases, VLDB, pp. 536–547 (2003)
3. Haux, R., Eckert, U.: Non deterministic dependencies in relations: an extension of the concept of functional dependency. Inf. Syst. **2**(10), 139–148 (1985)
4. Gupta, M., Gap, J., Aggarwal, C., Han, J.: Outlier detection for temporal data. Synth. Lect. Data Min. Knowl. Discov. **5**(1), 1–129 (2014)
5. Zhang, A., Song, S., Wang, J., Philip, Y.: Time series data cleaning: from anomaly detection to anomaly repairing. PVLDB **10**(10), 1046–1057 (2017)
6. El Chamie, M., Janak, D., Açıkmeşe, B.: Markov decision processes with sequential sensor measurements. Automatica **103**, 450–460 (2019)
7. Kolanowski, K., Swietlicka, A., Kapela, R., Pochmara, J., Rybarczyk, A.: Multisensor data fusion using Elman neural networks. Appl. Math. Comput. **319**, 236–244 (2018)
8. Song, S., Gao, Y., Wang, J.: Cleaning timestamps with temporal constraints. PVLDB **9**(10), 708–719 (2016)
9. Jordan, H., Owen, S.V., Arun, K.: Automatic anomaly detection in the cloud via statistical learning. CoRR abs/1704.07706 (2017)