A NEW DEFINITION FOR THE INFORMATION CONTENT OF DATABASES

Dan UNGUREANU[1]

În această lucrare am introdus o definiție originală pentru conținutul de informație dintr-o bază de date şi am investigat cum poate fi folosită aceasta nouă definiție pentru a analiza calitatea design-ului unei baze de date. Am prezentat şi o sinteză a abordărilor existente de folosire a teoriei informației pentru studiul bazelor de date.

In this paper we introduced a new original definition for the information content of a database and we investigated how this new definition can be used to analyze the quality of the design of a database. We also presented a synthesis of the existing approaches for using information theory in order to study the database domain.

Keywords: information content, relational databeses, normal forms

1. Introduction

As the volume of data that computers can process has increased over time, it has become extremely important to analyze and improve the quality of databases.

A theory that describes how the data from a database can be stored, organized and accessed is named a database model. Several database models have been introduced over time but the one that is predominantly used today and which has been extensively analyzed is the relational model - initially introduced in [1].

Information Theory has been introduced by Claude Shannon in the classic paper [2] and it defines fundamental limits on signal processing operations - like storing, compressing and communicating data. It has since been extended for applications in various areas.

Different attempts to apply information theory to the database domain have been made over time. However most solutions proposed so far focus on specific cases, and the database community doesn't have a clear, generally accepted definition for the information carried by the data from a database.

In this paper we will propose a new way of reasoning about the "information content" of a database. We hope that this might be a building block

[1] PhD student, Faculty of Automatic Control and Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: ungureanud@gmail.com

towards the ultimate goal in this research area - to find a unified theory linking the databases domain and information theory (if it is even possible). We will then focus on how our definition of the information content can be used to analyze the quality of the design of a database.

We will also briefly present some of the main previous approaches in this area.

2. Overview and notations for relational databases

We represent with S a database schema that consists of a set of relation names $\{R_1, R_2, ..., R_n\}$. A relation name R has a set of m elements called attributes denoted with $attr(R) = \{A_1, A_2, ..., A_m\}$. Each attribute has a domain – the set of all the possible values the attribute can take.

An instance I for a database schema S is represented by a relation I(R) for each relation name R. A relation I(R) for a relation name R with m attributes is a finite set of m-tuples over the cross-product of the domain for each attribute. We use the notation |I(R)| for the number of m-tuples.

We define the size of I(R) as $||I(R)|| = m \times |I(R)|$, and the size of I, $|| I ||$, as the sum of the sizes or each relation of the instance.

The set of positions in I is $Pos(I) = \{(R,p,A) \mid R \in S, p \in I(R), A \in attr(R)\}$. We have that $|Pos(I)| = || I ||$.

A database schema S may have a set of integrity constraints Σ (first-order sentences over S). We also define the closure Σ+ as the set of all the constraints that are logically implied by a set of integrity constrains Σ, and we represent with (S, Σ) as the set of all database instances of S that satisfy Σ.

One widely used type of integrity constraints is the functional dependency (FD) between two sets of attributes from a relation name, for which the standard notation X -> Y is used. A functional dependency X->Y means that each value of X in the database has associated exactly one value of Y.

A FD is trivial if $Y \subset X$. X is a key if X -> attr(R) is in Σ+. A key X is said to be a candidate (or minimal) key if there doesn't exist any proper subset $X_1 \subset X$ such that X_1 is also a key. An attribute that belongs to a candidate key is said to be a prime attribute.

The Armstrong axioms for functional dependencies, initially introduced in [3], are:

1) If $Y \subset X$ then X -> Y (reflexivity)
2) If X -> Y then XZ -> YZ (augmentation)
3) If X -> Y and Y -> Z then X -> Z (transitivity)

In order to characterize "well"-designed schemata some sets of conditions for the data dependencies called *normal forms* have been introduced. *Normalization* has been defined as the transformation of a database schema with a

given set of dependencies into a well-designed schema that represents the same information but for which the data dependencies respect some conditions associated to a given normal form.

A database schema S with a set of functional dependencies Σ is said to be in BCNF (Boyce - Codd Normal Form) if for every nontrivial FD X -> Y we have that X is a key.

A database schema S is said to be in 3NF (3rd Normal Form) if for every nontrivial FD X->Y we have that either X is a key or every attribute from Y-X is a prime attribute.

3. A new definition for the information content in a database

We consider that a database is a carrier of information from a source of information (the real world) to a receiver of information (a user that will retrieve data stored in the database).

The problem of how much information is contained in a database has been approached over time at different database levels - the information capacity of a database schema, the information content of a database instance, of a relation, of an attribute or of a single position from a database instance.

We will propose a new definition for *the information content in a position from a database instance* – with respect to the previous knowledge the user has about the domain.

We think that the previous knowledge of the user regarding the domain of the database (the real world objects whose properties are stored in the database) should be taken into consideration when we talk about the information contained in a database. Integrity constraints for the database can be considered to be a kind of previous knowledge of the domain.

We analyze first the case when the user has no previous knowledge of the domain.

Our opinion is that in the general case the user doesn't want to identify particular objects from a set, but that knowing each value of each attribute may be useful for the user - because it describes one property of a real world object. Since there are no integrity constraints and the user doesn't have any previous knowledge about the domain of the database, the user cannot deduce some of the values from other values, or from his previous knowledge. We say then that each value for each attribute from a database carries 1 "unit" of information. In other words, the "information content" of each position of an instance of a database is 1.

When the user has however some previous knowledge about the domain of the database, the positions from the database may not all have the same information content.

We analyze next the general case of functional dependencies as a kind of previous knowledge of the database domain.

Let S be a database schema and let Σ be a set of functional dependencies. Using the Armstrong axioms for FDs we can assume that all the FDs we work with have only one attribute in the consequent. Let $I \in$ instances(S, Σ) be an instance of S and let p be a position from Pos(I).

We present next our definition of the information content IC(p) of a position p from an instance I with respect to a set of functional dependencies Σ. If p is a position from Pos(I) - that means it has the form (R, t, A) with $R \in S$, $t \in I(R)$, $A \in attr(R)$. Let m be the number of attributes from R and let r be the number of m-tuples from R.

Our definition is based on the following two cases:

- the value stored in the instance in position p cannot be deduced from the rest of the values in the instance using the functional dependencies. In this case we say that there is no redundancy in the position and we say that IC(p) = 1.
- the value stored in the instance in position p can be deduced from the rest of the values in the instance using the functional dependencies. It means that A appears in the consequent of at least a FD, and that this dictates that the value that it takes for position p must be equal to the value of the same attribute A in a number of $k > 0$ other m-tuples from R (except t). We then define IC(p) = 1 / (k+1).

So in the general case

$$IC(p) = 1 / (k+1)$$

where k is the number of other tuples which must have the same value for the attribute of position p due to the FDs.

All the positions for the attribute in those tuples combined contain only one unit of information (the common value of the attribute). We say that the unit of information is divided between the positions.

We observe that when $k>0$, if the value is lost it can be obtained from the other positions of the instance, due to the integrity constraints Σ. We can say there is redundancy in the database and IC(p) will have a value less than 1.

A pseudo-code - type computation of the information content for a position p is presented below.

(1) *if* A does not appear in the consequent of any non-trivial FD *then*
    (2) IC(p) = 1
(3) *else*

```
/* If A appears in the consequent of at least one non-trivial FD */
    (4) if r = 1 then
        (5) IC(p) = 1
    (6) else
      /* If r > 1 we determine how many of the rest of (r-1) m-tuples from R
(except t - the tuple to which p belongs) must have the same value for attribute A
as the one in p, due to a FD */
        (7) identical_values = 0
        (8) for t1 in all the remaining (r-1) m-tuples
            (9) determines_p = false
            (10) for each non-trivial FD of the form X -> A
                (11) if the values of the attributes from X from t1 are all equal
to the corresponding values from t then
                    (12) determines_p = true
                (13) end if
            (14) end for
            (15) if determines_p then
                (16) identical_values++;
            (16) end if
        (17) end for
        (18) IC(p) = 1 / (1 + identical_values)
    (19) end if
(20) end if
```

We say that a database schema S with a set of functional dependencies Σ is well-designed if for any instance I and for any $p \in Pos(I)$ we have IC(p) = 1.

We analyze next some properties of the new measure.

Theorem 1. If $\Sigma = 0$ then (S, Σ) is well designed.

This follows from the definition: no value from any position can be deduced from the other positions because there are no constraints, so IC(p) = 1 for all positions p.

Theorem 2. (S, Σ) is well designed if and only if it is in BCNF.

(=>) We have that (S, Σ) is well designed and we assume that it is not in BCNF.

Then there exists a relation $R \in S$ and X -> A a nontrivial FD from Σ such that $X, A \subset attr(R)$ and X is not a key in R. This means that there is at least one attribute $B \in attr(R) - X - A$.

We consider an instance I with only two tuples that have the same values in the attributes from X (this must be possible because X is not a key). Then the

tuples must also have the same values for A due to the FD, so IC(p) = 0.5, which contradicts the fact that (S, Σ) is well designed.

(<=) We have that (S, Σ) is in BCNF and we assume that it is not well designed.

Then there exists an instance I and a position p = (R, t, A) such that IC(p) < 1. From the definition of the IC measure we see that IC(p) < 1 means that there must exist:

- a m-tuple from I(R) other than t
- an associated non-trivial FD of the form X -> A

such that all the values of the attributes from X correspond in the 2 m-tuples. But since (S, Σ) is in BCNF it means that X must be a key, so it cannot contain identical values for all its attributes in 2 different m-tuples. We have thus obtained a contradiction, which proves the initial implication.

We define the *information content percentage* (ICP) of a database instance I as the sum of information content of all the positions p divided by the total number of positions |Pos(I)| = || I ||.

Similarly we define the *information content percentage* (ICP) of a relation I(R) as the sum of information content of all the positions p from the relation divided by the total number of positions |Pos(I(R))| = || I(R) ||.

From Theorem 2 we can then deduce that a database schema with a set of functional dependencies (S, Σ) is in BCNF if and only if ICP(I) = 1 for every instance I. We can also conclude that by using the BCNF normalization algorithm for an instance of the database schema, the ICP value of the instance doesn't decrease (because after the normalization has been completed, the ICP of the resulting instance is 1, while the ICP of the initial instance is at most 1).

4. Previous uses of information theory in the databases domain

The main concept used in information theory, introduced in [2], is that of entropy - the amount of information provided by a certain event.

If an event X can have a finite number n of different outcomes { X_1, X_2, …, X_n}, each with probability $p(X_i)$, 1<=i<=n, than the average amount of information provided by the fact that a particular value of the event has occurred, denoted by H(X), is defined to be $H(X) = \Sigma (- p(X_i)\log(p(X_i)))$.

Since some of the probabilities Pi can be 0, by convention 0 x log(0) = 0.

Over time there have been approaches to analyze the information content at diferent levels in a database: at the level of a database schema, at the level of attributes from a relation, at the level of a position from a database instance.

In [4] a measure called 'information content' was defined for the attributes selected for a relation scheme. The value of the information content was used to rank the attributes - at the stage of logical design a database, before populating it

with data. This may help the designer of the database to decide about which attributes should be included in a database or to compare different relation schemes. The analysis of the attributes is accomplished by using data from simulations or from preliminary observations on the modeled system.

For a given attribute when the values from a finite set of tuples are considered, the entropy of the attribute is the average information contained in them. For an attribute A_j which has k different values in a finite set of n tuples, each value appearing with the frequency ti, the entropy of the attribute is defined as $H_{Aj} = -(t_1 / n).lg(t_1/n) – (t_2/n).lg(t_2/n) - … - (t_k/n).lg(t_k/n)$. This corresponds to the Shannon entropy where the tuples are viewed as temporal events occurring when, for example, one of the attributes takes certain values on a given set.

The information gain associated with an attribute A_j is $G(A_j) = H(A_j) – H_{avg}$, where H_{avg} is the average information for the relation schemes that contain the attribute A_j.

In [5] entropy was used as an information metric to quantify the information associated with a set of attributes, but this time for actual instances of the databases, and not in the stage of conceptual design of the database. It was shown that functional and multivalued dependencies can be expressed using entropies, and that their inference rules can be proven in terms of entropies.

In [6] the information content of data from the database has been defined as the instance of the database and the information capacity of a data schema has been defined as the collection of instances of the data schema. Four definitions of "dominance" between pairs of relational database schemata were given, resulting in measures of relative information capacity between schemata.

Another investigative direction has been based on the definition of the informational content of a signal (or structure) introduced in [7]:

A signal *r* carries the information that *s* is *F* = The conditional probability of *s*'s being *F*, given *r* (and *k*), is 1 (but, given *k* alone, less than 1).

Here *k* is a variable that takes into account how what an agent already knows can determine the information that a signal carries to the agent. It was also argued that the entropy from [2] defines the amount of information for a collection of messages, but it can't define the information content of a single variable. In [7] a random event was also informally called state of affairs.

In [8] the authors expanded on this idea and considered that data in a database may be seen as a type of signals. They defined the data in terms of random variables, random events and particulars of random events. The paper defined the information content of a state of affairs and it also proposed a definition of the information content inclusion relation (IIR) between two events:

- if X and Y are two random events, there is an IIR from X to Y if every possible particular of Y is in the information content of at least one particular from the random event X

Then the information content of a random event X was defined as all the random events with which X has an information content inclusion relation. The authors than introduced inference rules for IIR, analyzed their properties and gave an example of how they could be used in a database setting.

The authors of [9] expanded on the research direction from [8]. They defined the closure of a set of IIRs as all the IIRs that can be logically derived from them (a notion similar to the closure of a set of integrity constraints) and also the IIR closure of a random event as all the random events that can be derived from it using IIR inference rules starting from an initial set of IIRs.

In [10] an information-theoretic measure of the *information content* associated to a position of an instance of a database with respect to a given set of constraints was introduced. The measure, called relative information content, will be briefly described in this section.

The set of all the possible values for each attribute (the domain of the attribute) was considered to be the set of positive integers. The active domain of a database instance I was defined as the set of all the positive integers that occur in I. Since entropy can be defined only for events with a finite number of possible outcomes, the set $inst_k(S, \Sigma)$ was defined to be the subset of the total instances for which the active domain is in the range [1,k]. For all positive integers k a measure was defined of the relative information content of a position $INF_k(p \mid \Sigma)$ that works for instances I from $inst_k(S, \Sigma)$.

Basically, the goal was to measure how much the value in a position p is determined by any set of other positions from I. For any random subset X of Pos(I) - {p} the assumption is made that the values in those positions from X are lost and then they are restored randomly from the interval [1, k]. This action produces an information about the value of position p (also taken from the [1, k] interval). The average of this information for the possible subsets X represents $INF_k(p \mid \Sigma)$.

Such a position can theoretically have k different values of the corresponding attribute (but not all of them may lead to a database instance that satisfies the integrity constraints). The maximum value of entropy for an event that can have k different values is known to be log k. The general measure INF (p | Σ) was then defined as the limit of the ratio $INF_k(p \mid \Sigma) / \log(k)$, when $k \to \infty$.

A database schema (S, Σ) was defined to be well-designed if for every instance I and every position p Pos(I), INF (p | Σ) = 1. Also it was showed that for the case when Σ consists only of FDs, a schema (S, Σ) is well-designed if and only if it is in BCNF.

In [11] the measure introduced in [10] was used to analyze worst-case redundancy for databases. It was remarked that using BCNF has the disadvantage that after normalization some of the functional dependencies valid in the initial

database may be lost. That is the reason that 3NF is preferred in a lot of cases in practice.

A measure called the *guaranteed information content* for a schema with integrity constraints expressed by functional dependencies was used to define the lowest information content that could be found for a position in any instance of the schema that respects the constraints. The notion of guaranteed information content was also used to characterize relation schemas which respect a certain condition C (that can be used to specify a normal form), and which have a given number of attributes m.

The *price of dependency preservation* for a normal form was used to define the lowest amount of redundancy that is present in relations schemas for the normal form. It was then proved that using these measures based on the relative information content, 3NF has the lowest price of dependency preservation compared to all the dependency preserving normal forms.

5. Conclusions

We have analyzed in this paper the application of information theory to the database domain. We have introduced a new definition for the information content of a position in a database. The new definition has been used to characterize BCNF. Also a new measure was introduced – the information content percentage for a database instance, which was used to reason about the BCNF normalization algorithm at instance level.

We have also briefly presented previous research in this area, where the information content has been studied at different levels in a database.

The next step of the research is to characterize 3NF using this new definition and to analyze the 3NF normalization algorithms using the information content percentage measure.

We also want to analyze how the new definition we introduced can be linked with the semantic theory of information introduced in [7].

R E F E R E N C E S

[1] *E.F. Codd*, "A Relational Model of Data for Large Shared Data Banks", in Communications of the ACM, **vol. 13**, no. 6, June 1970, pp. 377-387

[2] *C. E. Shannon*, "A Mathematical Theory of Communication", in Bell System Technical Journal, **vol. 27**, July and Oct. 1948, pp. 379-423 and 623-656

[3] *William Ward Armstrong*, "Dependency Structures of Data Base Relationships", in IFIP Congress, pp. 580-583, 1974

[4] *Mircea Petrescu*, "Information theory aspects in relational database design", in Advances in Electrical and Computer Engineering Journal, **vol. 2**, issue 1, 2002

[5] *Tony T. Lee*, "An Information-Theoretic Analysis of Relational Databases - Part I: Data Dependencies and Information Metric", in IEEE Transactions on Software Engineering, **vol. SE-13**, no. 10, Oct. 1987, pp. 1049-1061

[6] *Richard Hull*,"Relative information capacity of simple relational database schemata", in SIAM Journal of Computing, **vol. 15**, no. 3, Aug. 1986, pp. 856-886
[7] *Fred Dretske*, Knowledge and the Flow of Information, MIT Press, 1983
[8] *K. Xu, J. Feng, M. Crowe*, "Defining the notion of 'Information Content' and reasoning about it in a database", in Journal of Knowledge and Information Systems, **vol.18**, issue 1, January 2009
[9] *J. Feng, D. Salt*, "Information Content Inclusion Relation and its Use in Database Queries", in Journal of Software Engineering and Applications, **vol. 3**, no. 7, Mar. 2010, pp. 255-267
[10] *M. Arenas, L. Libkin*, "An Information-Theoretic Approach to Normal Forms for Relational and XML Data", in Journal of the ACM, **vol. 52**, issue 2, Mar. 2005, pp. 246-283
[11] *S. Kolahi, L. Libkin,* "An information-theoretic analysis of worst-case redundancy in database design", in ACM Transactions on Database Systems, **vol. 35**, issue 1, 2010