МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2006
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2006
Proceedings of the Thirty Fifth Spring Conference of
the Union of Bulgarian Mathematicians
Borovets, April 5–8, 2006

FINDING THE BRANCHING DEPENDENCIES IN RANDOM DATABASES*

Tsvetanka L. Georgieva

In the present paper some properties of the branching dependencies are examined. We define a minimal branching dependency and we propose an algorithm for finding all minimal branching dependencies between a given set of attributes and a given attribute in relation of random database. A realization of the presented algorithm by using a data cube is described.

1. Introduction. Functional dependencies are relationships between attributes of a database relation. Some functional dependencies are defined during the process of the database design and they are used to support the referential integrity. But the constraints are few and often too general in sense that they are valid in all possible database states. The discovery of the functional dependencies that reflect the present content of the relation is an important database analysis technique. The basic motivation for discovery of the functional dependencies, which hold in the current instance of a relation, is discovery of valuable knowledge of the structure of the relation instance.

In some cases, a given functional dependency may not hold only for a few tuples. This functional dependency can be thought as approximate, i.e. almost holds. Approximate functional dependencies also represent valuable knowledge of the structure of the current instance of the relation. The discovery of such knowledge can be valuable for analyzing the data that is contained in the database.

The functional dependency requires the values in a given set of attributes to uniquely determine the value in a given attribute. In [3] a branching dependency that is a more general dependency, than the functional dependency is introduced. The paper [4] contains investigations concerning this dependency. The branching dependency allows determining the maximal number of the different values in a given attribute corresponding of one or more different values in a given set of attributes in the relation. This knowledge is additional information that may be useful for analyzing the current content of the database.

In the present paper a minimal branching dependency is defined and some properties of the branching dependencies are examined. An algorithm for finding all minimal branching dependencies between a given set of attributes and a given attribute is proposed. A database relation with unknown structure is considered. The values of the tuples in the

*Supported partially by the Bulgarian National Science Fund under Grant IO-03/2005.

attributes are randomly generated. A realization of the represented algorithm by using a data cube is described.

The rest of the paper is organized as follows. In section 2 we present a brief survey on the related work. In section 3 the task for discovering all minimal branching dependencies is formulated. In section 4 the dependencies between attributes are analyzed with the purpose of revealing some valuable knowledge of the data that is contained in the database. In section 5 an algorithm for finding all minimal branching dependencies is described. Section 6 represents some details about the realization of the proposed algorithm and its comparison with other algorithms is exposed in section 7. Section 8 gives the conclusion of this paper.

2. Related work. The basic definitions and properties connected with the functional dependencies in relational databases are represented in detail in [5,8]. The functional dependencies in random databases are investigated in [2]. In [7] an efficient algorithm for finding functional dependencies from large databases is presented. This algorithm is based on partitioning the set of tuples with respect to their attributes values. In [6] an approach to visualizing functional and approximate dependencies that allows revealing some characteristics of attributes and their local structure is described.

Some theorems valid for functional dependencies are generalized to branching dependencies in [3]. Moreover some implications among branching dependencies are investigated. In [4] the estimations for the minimal number of tuples in a relation that results the sets of attributes, which (p, q)-depend on sets A of attributes are found.

In the present paper the task for finding all branching dependencies between a given set$A$ of attributes and a given attribute b is considered. For that purpose a minimum branching dependency is defined in a manner that the validity of all branching dependencies between A and b can be established if all minimal branching dependencies are known. Moreover some properties of the branching dependencies are examined that allow to prune some values of p and q during the search of the branching dependencies between A and b and to create an efficient algorithm.

3. Functional and Branching Dependencies. Let R be a database relation and let Ω be the set of attributes of the relation R. The number of the attributes is $|\Omega| = n$. We say that the *functional dependency* (*FD*) *holds* or *is valid* in R, if for any two tuples r, $s \in$ we have: if $r(a_k) = s(a_k)$ for all $a_k \in A$, $A = \{a_1, \ldots, a_i\}$, $A \subset \Omega$, $k = 1, \ldots, i$, then $r(b) = s(b)$ for $b \in \Omega$. We also say that b functionally depends on A and write $A \to b$. The functional dependency $A \quad \to \quad b$ is called *nontrivial* if $b \notin A$. We say that the functional dependency $A \to b$ is *minimal* if b is not functionally dependent on any subset of A, i.e. if $B \to b$ does not hold in R for any $B \subset A$.

Let π be the projection operator, δ be the duplicate-elimination operator and let $A \subset \Omega$, $b \in \Omega$, $b \notin A$ and $1 \leq p \leq q$, integers. We say that b (p, q)-*depends* on A if there are no $q + 1$ tuples such that they contain at most p different values in each $a_k \in A$, $k = 1, \ldots, i$, but $q+1$ different values in b. We also say that (p, q)-branching dependency holds and write $A \xrightarrow{(p,q)} b$.

Consequently the branching dependency $A \xrightarrow{(1,1)} b$ is FD $A \to b$. It is easy to see that if the branching dependency $A \xrightarrow{(p,q)} b$ holds, then the branching dependency $A \xrightarrow{(p,q_1)} b$ holds for each $q_1 > q$. Moreover if b (p, q)-depends on A for $q = |\delta(\pi_b(R))|$, then $A \xrightarrow{(p_1,q)} b$

holds for each $p_1 > p$. These conclusions give a reason for defining a minimum branching dependency.

We say that the branching dependency $A \overset{(p,q)}{\longrightarrow} b$ for $1 \leq p \leq q$ is *minimal* if for $q < |\delta(\pi_b(R))|$ the dependency $A \overset{(p,q_1)}{\longrightarrow} b$ does not hold for each $q_1 < q$, $p \leq |\delta(\pi_{a_1,\ldots,a_i}(R))|$ and for $q = |\delta(\pi_b(R))|$ the dependency $A \overset{(p_1,q)}{\longrightarrow} b$ does not hold for each $p_1 < p$, where p, q, p_1, q_1 are integers.

The central task we consider is with a given set A of attributes and an attribute$b$, find all minimal branching dependencies between A and b.

Proposition 1. *If $p = q$ and $|\delta(\pi_b(R))| > q$, then the branching dependency $A \overset{(p,q)}{\longrightarrow} b$ holds if and only if the functional dependency $A \rightarrow b$ is valid.*

Proof. Let $A \overset{(p,p)}{\longrightarrow} b$ be a valid branching dependency. We assume that the FD $A \rightarrow b$ does not hold, i.e. there are two different tuples r and s, such that $r(a_k) = s(a_k)$ for all $a_k \in A$, $k = 1, \ldots, i$ and $r(b) \neq s(b)$. Since the number of the different values in b is at least $p + 1$, then besides r and s there are at least another $p - 1$ tuples, which contain different values in the attribute b. We suppose that r_l for $l = 1, \ldots, p-1$ are these tuples. Consequently r, s, $\{r_l\}_{l=1,\ldots,p-1}$ are $p + 1$ tuples with at most p different values in the attributes of A, but $p + 1$ different values in the attribute b. This conclusion contradicts of the assumption that the branching dependency $A \overset{(p,p)}{\longrightarrow} b$ holds. Hence we can make the conclusion that if $r(a_k) = s(a_k)$ for all $a_k \in A$, $k = 1, \ldots, i$, then $r(b) = s(b)$, i.e. the FD $A \rightarrow b$ is valid.

If the FD $A \rightarrow b$ is valid, the values in the attributes of A uniquely determine the value in the attribute b and therefore it is not possible to exist $p+1$ tuples with at most p different values in the attributes of A, but $p + 1$ different values in the attribute b.

Proposition 2. *If the branching dependency $A \overset{(p,q)}{\longrightarrow} b$ does not hold and $|\delta(\pi_b(R))| > q + 1$, i.e. there are at least $q + 2$ different values in the attribute b, then the branching dependency $A \overset{(p+1,q+1)}{\longrightarrow} b$ also does not hold.*

Proof. From the supposition that the branching dependency $A \overset{(p,q)}{\longrightarrow} b$ does not hold, we can make the conclusion that there are $q + 1$ tuples $r_1, \ldots, r_{q+1}$, such that they have at most p different values in the attributes in A and $r_{j_1}(b) \neq r_{j_2}(b)$ for each $j_1 \neq j_2$, $1 \leq j_1, j_2 \leq q + 1$. Since there exist at least $q + 2$ different values in the attribute b, let $r_l(b) \neq r_j(b)$ for $j = 1, \ldots, q + 1$. Then the tuples $r_1, \ldots, r_{q+1}$, r_l are $q + 2$ in number tuples, which have at most $p + 1$ different values in A and $q + 2$ different values in the attribute b. Consequently the branching dependency $A \overset{(p+1,q+1)}{\longrightarrow} b$ does not hold.

Corollary 2.1. *If the branching dependency $A \overset{(1,q)}{\longrightarrow} b$ does not hold and $|\delta(\pi_b(R))| > q + 1$, i.e. there are at least $q + 2$ different values in the attribute b, then the branching dependency $A \overset{(2,q+1)}{\longrightarrow} b$ does not hold.*

If first we find the minimal branching dependency for $p = 1$, i.e. $A \overset{(1,q)}{\longrightarrow} b$, then since the branching dependency $A \overset{(1,q-1)}{\longrightarrow} b$ does not hold, from Corollary 2.1 follows that the branching dependency $A \overset{(2,q)}{\longrightarrow} b$ also does not hold. Therefore the next step

should be testing the validity of the branching dependency $A \overset{(2,q+1)}{\longrightarrow} b$, and so on, while $p \leq |\delta(\pi_{a_1,\ldots,a_i}(R))|$ and $q \leq |\delta(\pi_b(R))|$.

4. Analyzing the Dependencies between Attributes in Random Databases. One of the ways of defining the approximate dependency is based on the minimal number of tuples that need to be removed from the relation R for relevant FD $A \rightarrow b$ to hold in R [7]. The error $e(A \rightarrow b)$ is defined as $e(A \rightarrow b) = \min\{|S| \text{ for } S \subseteq R \text{ and } A \rightarrow b \text{ holds in } R \backslash S\}/|R|$. With a given value ε such that $0 \leq \varepsilon \leq 1$ we say that $A \rightarrow b$ is an *approximate* (*functional*) *dependency* if $e(A \rightarrow b)$ is at most ε.

Let σ be the selection operator and $v = (v_1, \ldots, v_i)$ be any element of the set $\delta(\pi_{a_1,\ldots,a_i}(R))$. The error can be calculated by the following expression:

$$e(A \rightarrow b) = 1 - \sum_{v \in \delta(\pi_{a_1,\ldots,a_i}(R))} \max\{|\pi_{b_k}(\sigma_{A=v}(R))| \text{ for } b_k \in \delta(\pi_b(\sigma_{A=v}(R)))\}/|R|.$$

If for an arbitrary element $v \in \delta(\pi_{a_1,\ldots,a_i}(R))$ the corresponding value b_v of the attribute b is different from $b_{frequent}$, which is one of the values satisfying the condition

$$|\pi_{b_{frequent}}(\sigma_{A=v}(R))| = \max\{|\pi_{b_k}(\sigma_{A=v}(R))| \; for \; b_k \in \delta(\pi_b(\sigma_{A=v}(R)))\},$$

b_v is considered as exception.

When an approximate dependency $A \rightarrow b$ with $e(A \rightarrow b) \leq \varepsilon$ is found, it is interesting to establish whether the exceptions refer to only one or few in number different values of A or the exceptions occur for mostly different values of A. This information may be used for choosing one of the following two approaches to correction of the error:

- deletion of minimal number of tuples in the relation R for the FD $A \rightarrow b$ to hold in R, i.e. ignoring the noise;
- addition of minimal number of attributes in A to obtain the set of attributes A', such that the FD $A' \rightarrow b$ already holds in R (in [1] an algorithm for finding such enlargement A' is proposed).

The first approach is appropriate if the exceptions refer to only one or few in number different values of A and they can be ignored by considering as noise. This case corresponds to a minimal branching dependency $A \overset{(1,q)}{\longrightarrow} b$ for q approximately equal to the number of the tuples in which the FD $A \rightarrow b$ is violated. The second approach is appropriate if the exceptions occur for mostly different values of A. Then we can make different conclusions according to the data itself. This case corresponds to a minimal branching dependency $A \overset{(1,q)}{\longrightarrow} b$ for q with comparatively small value.

When the error $e(A \rightarrow b)$ is not small according to the sense into the concrete data the analogue two cases are of interest, i.e. to establish whether the "exceptions" refer to few in number different values or they occur for mostly different values of A.

5. An Algorithm for Finding the Minimal Branching Dependencies. The algorithm represented as Algorithm 1 starts with validity testing of the functional dependency. If FD $A \rightarrow b$ does not hold it computes the error.

Algorithm 1

Input: relation R with a set of attributes Ω; chosen set A of attributes, $A \subset \Omega$, $A = \{a_1, \ldots, a_i\}$; an attribute$b$, $b \in \Omega$, $b \notin A$

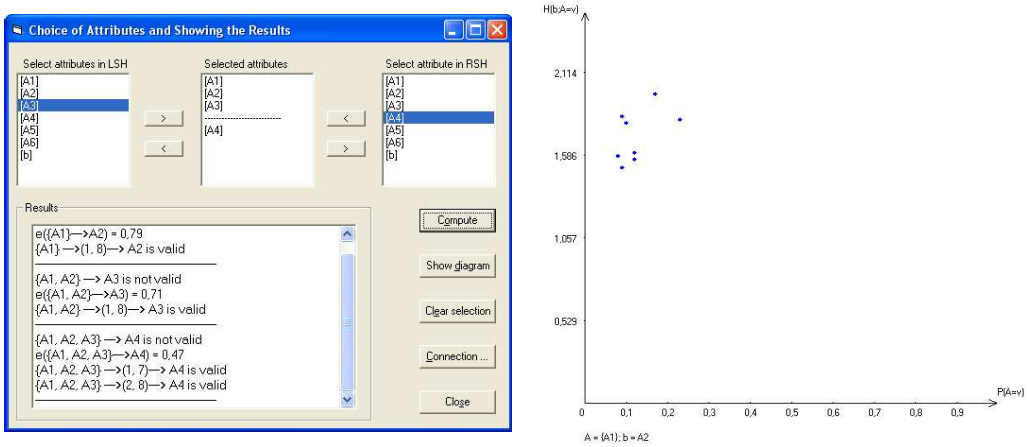
Output: all minimal branching dependencies between A and b

1. **If** $|\delta(\pi_{a_1,\ldots,a_i}(R))| = |\delta(\pi_{a_1,\ldots,a_i,b}(R))|$
   2. **Output** FD $A \rightarrow b$
3. **Else**
   4. **Output** the error $e(A \rightarrow b)$
   5. $q = \max\{|\delta(\pi_b(\sigma_{a_1=v_1 \text{ and } \ldots \text{ and } a_i=v_i}(R)))| \; for \; (v_1,\ldots,v_i) \in \delta(\pi_{a_1,\ldots,a_i}(R))\}$
   6. **Output** $A \xrightarrow{(1,q)} b$
   7. $p = 2$
   8. $q = q + 1$
   9. **While** $p \leq |\delta(\pi_{a_1,\ldots,a_i}(R))|$ and $q \leq |\delta(\pi_b(R))|$
      10. $q = \max\{|\delta(\pi_b(\sigma_C(R)))|$, *where* C *is the condition*
          $(a_1 = v_{11}$ AND $a_2 = v_{21}$ AND $\ldots$ AND $a_i = v_{i1})$
          OR $(a_1 = v_{12}$ AND $a_2 = v_{22}$ AND $\ldots$ AND $a_i = v_{i2})$
          OR $\ldots$ OR $(a_1 = v_{1p}$ AND $a_2 = v_{2p}$ AND $\ldots$ AND $a_i = v_{ip})$
          for $(v_{1j},\ldots,v_{ij}) \in \delta(\pi_{a_1,\ldots,a_i}(R))$, $j = 1,\ldots,p\}$
      11. **Output** $A \xrightarrow{(p,q)} b$
      12. $p = p + 1$
      13. $q = q + 1$

On line 5 the algorithm finds the maximal number of the different values in the attribute b that correspond to a value of A. By this way it finds the minimal branching $A \xrightarrow{(1,q)} b$ dependency. If the number of the different values in b is at most q, i.e. $|\delta(\pi_b(M))| \leq q$, then the branching dependency $A \xrightarrow{(p,q)} b$ is always valid. Therefore the algorithm considers these values of q, which are less than $|\delta(\pi_b(M))|$ and for $q = |\delta(\pi_b(M))|$ finds the minimal p, for which the branching dependency $A \xrightarrow{(p,q)} b$ holds. On line 10 the algorithm finds the maximal number of the different values in the attribute b, that correspond to p in number different values of A and by this way it discovers the minimal branching $A \xrightarrow{(p,q)} b$ dependency.

6. Realization of the Algorithm for Finding the Minimal Branching Dependencies in Random Databases. The proposed approach uses a data cube with dimensions corresponding of all the attributes in the relation R and a measure whose value is obtained by computing the count of the tuples in R with the different values in the attributes. The measure of the data cube is computed with grouping by all possible combinations of attributes in R.

By means of graphical interface the realization of the algorithm allows to select the data cube, the attributes in the left-hand side (LSH) and the attribute in the right-hand side (RSH) of the dependency liable to analyzing (Fig. 1(a)). The application provides the possibility of additional analyzing through the proposed in [6] graphical presentation of the different values of $P(A = v) = |\sigma_{a_1=v_1 \text{ and } a_2=v_2 \text{ and } \ldots \text{ and } a_i=v_i}(R)|/|R|$ and relevant values of $H_b(\sigma_{A=v}(R)) = \sum\limits_{t \in \delta(\pi_b(\sigma_{A=v}(R)))} P(b=t) \log \dfrac{1}{P(b=t)}$ in order to reveal some characteristics of the local structure (Fig. 1(b)).



(*a*) (*b*)

Fig. 1 Results for 1024 tuples in the relation

For the realization of the algorithm we use the languages MDX (*Multidimensional Expressions*) [10, 11, 12] and Visual Basic [9].

7. Comparison with Other Algorithms. The proposed in [7] algorithm for discovering functional and approximate dependencies is based on partitioning the set of tuples with respect to their attributes values. On line 1 the algorithm represented in the present paper tests the validity of the FD and if the FD is not valid, on line 4 the algorithm computes the error. For that purpose we use a data cube in which previously found different values of the attributes and the count of the tuples in the relation with relevant values of the attributes are stored. The aggregate values computed and stored in the data cube facilitate the verification of the validity of the FD $A \to b$ by finding the number of the different values in the attributes of the sets A and $A \cup \{b\}$, the computation of the error $e(A \to b)$, as well as the discovery of the minimal branching dependencies. Therefore the usage of a data cube in the represented algorithm is more effective than the usage of a relational table-based structure that requires multiple scans of the data.

According to our studying of the previously published practical algorithms for discovering dependencies until now such that finds branching dependencies is not proposed. For all executions of line 10 in the proposed Algorithm 1 the following number values are compared:

$$\left(\begin{array}{c} |\delta(\pi_{a_1,\ldots,a_i}(R))| \\ 2 \end{array} \right) + \cdots + \left(\begin{array}{c} |\delta(\pi_{a_1,\ldots,a_i}(R))| \\ p \end{array} \right),$$

where $2 < p < q \leq |\delta(\pi_b(R))| \Rightarrow$ at most $|\delta(\pi_b(R))| - 2$ in number values are summed.

8. Conclusion. Analyzing the dependencies between attributes existing in a given moment allows revealing the valuable knowledge of the structure of the current instance of a relation. In the present paper the task for discovering all minimal branching dependencies is considered. An algorithm for finding all minimal branching dependencies is described. A realization of the proposed algorithm is represented by using the previously created data cube in order to increase the effectivity of computing the values needed for

finding the branching dependencies. Currently, we are investigating the execution time of the algorithm with the growth of the number of the tuples, the growth of the number of the different values in the attributes and the growth of i – the number of the attributes.

REFERENCES

[1] G. Bogdanova, T. Georgieva. Finding the Error-Correcting Functional Dependency by Using the Fractal Dimension, In *Proceedings of the Fourth International Workshop on Optimal Codes and Related Topics*, 2005, 20–26.

[2] J. Demetrovics, G. O. H. Katona, D. Miklós, O. Seleznjev, B. Thalheim. Functional Dependencies in Random Databases, *Studia Sci. Math. Hungar.*, **34** (1998), 127–140.

[3] J. Demetrovics, G. O. H. Katona, A. Sali. The Characterization of Branching Dependencies, *Discrete Applied Mathematics*, **40** (1992), 139–153.

[4] J. Demetrovics, G. O. H. Katona, A. Sali. Minimal Representations of Branching Dependencies, *Acta Sci. Math. (Szeged)*, **60** (1995), 213–223.

[5] H. Garcia-Molina, J. D. Ullman, J. Widom. Database Systems: The Complete Book, Williams, 2002.

[6] D. P. Groth, E. L. Robertson. An Entropy-based Approach to Visualizing Database Structure, In *Proceedings of the Sixth Working Conference on Visual Database Systems*, 2002, 157–170.

[7] Y. Huhtala, J. Karkkainen, P. Porkka, H. Toivonen. Tane: An Efficient Algorithm for Discovering Functional and Approximate Dependencies, *The Computer Journal*, **42** (1999), No 2, 100–111.

[8] J. Peneva. Databases – I part, Regalia 6, 2004 (in Bulgarian).

[9] W. Wang. Visual Basic 6: A Beginner's Guide, AlexSoft, 2002 (in Bulgarian).

[10] `http://www.georgehernandez.com/xDatabases/MD/MDX.htm`

[11] `http://www.microsoft.com/data/oledb/olap`

[12] `http://www.microsoft.com/sql`

Tsvetanka L. Georgieva
"St. St. Cyril and Methodius" University of Veliko Tarnovo
Department of Information Technologies
Veliko Tarnovo, Bulgaria
e-mail: cv.georgieva@uni-vt.bg

НАМИРАНЕ НА BRANCHING ЗАВИСИМОСТИ В БАЗИ ОТ ДАННИ СЪС СЛУЧАЙНО ГЕНЕРИРАНИ СТОЙНОСТИ

Цветанка Л. Георгиева

В настоящата статия са разгледани и доказани някои свойства на branching зависимостите. Дефинирана е минимална branching зависимост и е предложен алгоритъм за намиране на всички минимални branching зависимости между дадено множество от атрибути и даден атрибут в релация на бази данни със случайно генерирани стойности. Описана е реализация на представения алгоритъм чрез използване на куб с данни.