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

DISCOVERING THE ERROR-CORRECTING FUNCTIONAL DEPENDENCIES IN CASE OF UNKNOWN ORIGINAL DEPENDENCIES

Galina Bogdanova[*], **Tsvetanka Georgieva**[**]

The discovery of a connection between a given functional dependency in a dataset and that of the dataset obtained after its transmission through a noisy channel is an important database task. In case of data mining only the received data are known and the aim is to make conclusions about the functional dependencies of the completely unknown original dataset. The present paper represents an algorithm for finding the error-correcting functional dependencies in this case. The proposed solution to this problem uses the approximate dependencies whose definition is based on the fractal dimension of the corresponding datasets.

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.

Let the set of attributes Ω of the relation M has size $|\Omega| = n$. We say that the functional dependency (FD) *holds* or *is valid* in M if for any two tuples r, $s \in M$ we have: if $r(a) = s(a)$ for all $a \in A$, $A \subset \Omega$, then $r(b) = \mathrm{s}(b)$ for $b \in \Omega$. We also say that b depends functionally on A and write $A \rightarrow b$. The FD $A \rightarrow b$ is called *nontrivial* if $b \notin A$. We say that the FD $A \rightarrow b$ is *minimal* if b is not functionally dependent on any subset of A, i.e. if $B \rightarrow b$ does not hold in M for any $B \subset A$.

A dataset which forms the present content of the relation can be seen as an $m \times n$ matrix or as a set of m points in an n-dimensional space. These data are transmitted through a noisy channel. M^* denotes the dataset of the data obtained after the transmission. In Model 1 examined in [4] it is assumed that the structure of M is known as well as the received rows of M^* and the aim is to make conclusions based on this information about the connection between the structure of M and this of M^*. It is supposed, e.g., that $A \rightarrow b$ $(A \subset \Omega,\ A = \{a_1, \ldots, a_i\},\ b \in \Omega)$ holds in M. Then, the data in a row in the columns of A determine the data of the same row in a column b. However, only corresponding rows in M^* are known. The data in columns of A in M^* do not necessarily

[*]Supported partially by the Bulgarian National Science Fund under Grant MM-1405/2004
[**]Supported partially by the Bulgarian National Science Fund under Grant IO-03/2005

determine the data in b, since these data may be distorted. Consequently, it is possible that $A \to b$ does not hold in M^*. The aim is to establish whether it is possible to enlarge A into an A', so that the data in A' in M^* already determine b, and if possible, to determine to what extent should it be enlarged. Let $A' = \{a_1, \ldots, a_i, x_1, \ldots, x_j\}$ in M^* denote the set so that $A' \to b$ holds, i.e. the data in A' uniquely determine the data in b in spite of the errors. This FD is called *error-correcting functional dependency.*

In Model 2 examined in [4] nothing is known about the functional dependencies in M. In this case of data mining only the received rows of M^* are known and the aim is to make conclusions about the functional dependencies of the completely unknown M.

In [4] inequalities between the sizes of the sets occurring in the functional dependencies in M and the error-correcting functional dependencies are found, considering both of the models. An algorithm for finding the error-correcting functional dependency with known original dependency by using the fractal dimension is proposed in [3].

The basic definitions and properties connected with the FDs in relational databases are represented in detail in [6, 10]. In [7] an efficient algorithm for finding FDs from large databases is presented. This algorithm is based on partitioning the set of rows with respect to their attributes values. Minimizing database access during the discovery of FDs and maintenance of the discovered FDs can be achieved by axiomatization of functional dependencies and independencies presented in [1].

In the present paper an algorithm for finding the error-correcting functional dependencies in case of unknown original dependencies is represented. The rest of the paper is organized as follows. Section 2 presents the main fractal properties. In section 3 an algorithm for finding the error-correcting functional dependencies in case of unknown original dependencies is described by using the fractal dimension. Section 4 presents an application of the algorithms for discovering the error-correcting dependencies for collaborative filtering. The proposed algorithm is analyzed in section 5 and it is compared with other algorithms in section 6. Section 7 is the conclusion of this paper.

2. Fractal dimension. A set of points is a fractal if it exhibits self-similarity over all scales. The fractals have been used in numerous disciplines [5, 8]. In the database area, e.g., the fractals have been successfully used to estimate the selectivity of spatial queries [2], quickly select the most important attributes for a given dataset [12].

The fractal sets are characterized by their fractal dimension. In fact, there is an infinite family of fractal dimensions. For a dataset with finite number of points, the generalized fractal dimension D_q is defined with (1).

$$D_q = \begin{cases} \dfrac{1}{q-1}\dfrac{\partial \log \sum_i C_{r,i}^q}{\partial \log r} & \text{for } \; q \neq 1 \\ \dfrac{\partial \log \sum_i C_{r,i} \log C_{r,i}}{\partial \log r} & \text{for } \; q = 1 \end{cases} \tag{1}$$

where the dataset is embedded in an n-dimensional grid which cells have sides of size r; $C_{r,i}$ is the frequency with which data points fall into the i-th cell; q is a real number.

Among the dimensions described by (1), the following dimensions are widely used:

- the Hausdorff fractal dimension D_0 obtained for $q = 0$;
- the correlation fractal dimension D_2 obtained for $q = 2$;
- the information fractal dimension D_1 obtained for $q = 1$.

The correlation dimension measures the probability that two points chosen at random will be within a certain distance of each other. Changes in the correlation dimension mean changes in the distribution of points in the dataset. Changes in the information dimension means changes in the entropy and, therefore, point to changes in trends.

Fast algorithms exist to compute these dimensions. Since the dataset is considered as a set of m points in an n-dimensional space, the algorithms for calculation of the fractal dimension use n-grid with grid-cells of side r. Let $C_{r,i}$ denote the number of points in each i-th cell. Then, the value of $S_2(r) = \sum_i C_{r,i}^2$is computed. The correlation fractal dimension is the derivative of $(\log S_2(r))$ with respect to the logarithm of the size r. Thus, the correlation fractal dimension D_2 of the dataset can be obtained by calculating the slope a of the line $y = ax + b$ that is the best approximation of $y_i = \log(S_2(r_i))$, $x_i = \log(r_i)$ for different values of the size r.

3. An Algorithm for discovering the error-correcting functional dependencies in case of unknown original dependencies. The task we consider is the following: given a dataset M^* obtained after the transmission of the unknown original dataset M, find all error-correcting functional dependencies. This problem can be solved by using the approximate functional dependency. An approximate functional dependency is a functional dependency that almost holds. There are different ways of defining the approximate dependency $A \to b$. The definition we use is based on the difference of the fractal dimension of the dataset $\pi_{a_1,\ldots,a_i}(M^*)$ and the fractal dimension of the dataset $\pi_{a_1,\ldots,a_i,b}(M^*)$, where π is the projection operator and$A = \{a_1, \ldots, a_i\}$. The fractal dimension we compute in our algorithms is correlation fractal dimension or information fractal dimension and, therefore, let $F(X)$ be the correlation or information fractal dimension of some dataset X. If $e(A \to b) = |F(\pi_{a_1,\ldots,a_i}(M^*)) - F(\pi_{a_1,\ldots,a_i,b}(M^*))|$, then we say that $A \to b$ is an *approximate (functional) dependency* if $e(A \to b)$ is at most ε, where ε is a given threshold.

The algorithm represented as Algorithm 1 starts with $C_1 = \{\{a\}|a \in \Omega\}$ and computes C_2, and so on, according to the information obtained during the algorithm. A set C_k consists of attribute sets of size k such that they can potentially be used to construct dependencies.

Algorithm 1

```
Input: dataset $M^*$ with the set of $n$ attributes $\Omega$; $b \in \Omega$; threshold $\varepsilon$
Output: minimal nontrivial approximate dependencies $A \to b$, $A \subset \Omega$
1) $C_1 = \{\{a\}|a \in \Omega\}$
2) $k = 1$
3) while $C_k \neq \emptyset$ and $k \geq 1$ and $k < n$
4)     Compute_dependencies($C_k$)
5)     $k = k + 1$
6)     $C_k = \{A|A$ is $k$-element subset of $\Omega$ and for each $(k-1)$-element subset $B$ of $A$ approximate FD $B \to b$ is not valid$\}$
```

The procedure Compute_dependencies(C_k) finds the minimal dependencies with the left-hand side in C_k.

Procedure Compute_dependencies(C_k)

```
1) for each $A \in C_k$ do
2)     if $A \to b$ is valid then
```

3) **output** $A \to b$

The validity testing on line 2 is based on the computation of $e(A \to b)$ and if $e(A \to b) \leq \varepsilon$ then the dependence $A \to b$ holds. The pre-defined value ε depends on how precise the resulting dataset M^* has preserved the characteristics of the original dataset M.

4. Using the error-correcting dependencies for collaborative filtering. The proposed algorithm can be used as a part of the process of collaborative filtering. The field of collaborative filtering attempts to automate the process of organizing and recommending information to the users by supporting the users in making decisions and finding a set of people who are likely to provide good recommendations for a given person.

The general idea of collaborative filtering is to find some groups of people (users) where each group ranked approximately the same items and gave approximately the same ranking to those items and then make recommendations to the active user by matching the active user's profile to each group. The quality of the recommendations can be influenced by many factors: sparsity – very few people rank exactly the same items; noisiness – people may not give ranks or may not give true ranks.

Collaborative filtering uses a database of user preferences to predict additional items that a new user, referred to as the active user, might like. The problem space of collaborative filtering can be formulated as a matrix of users versus items, with each cell representing a user's rating on specific item [11].

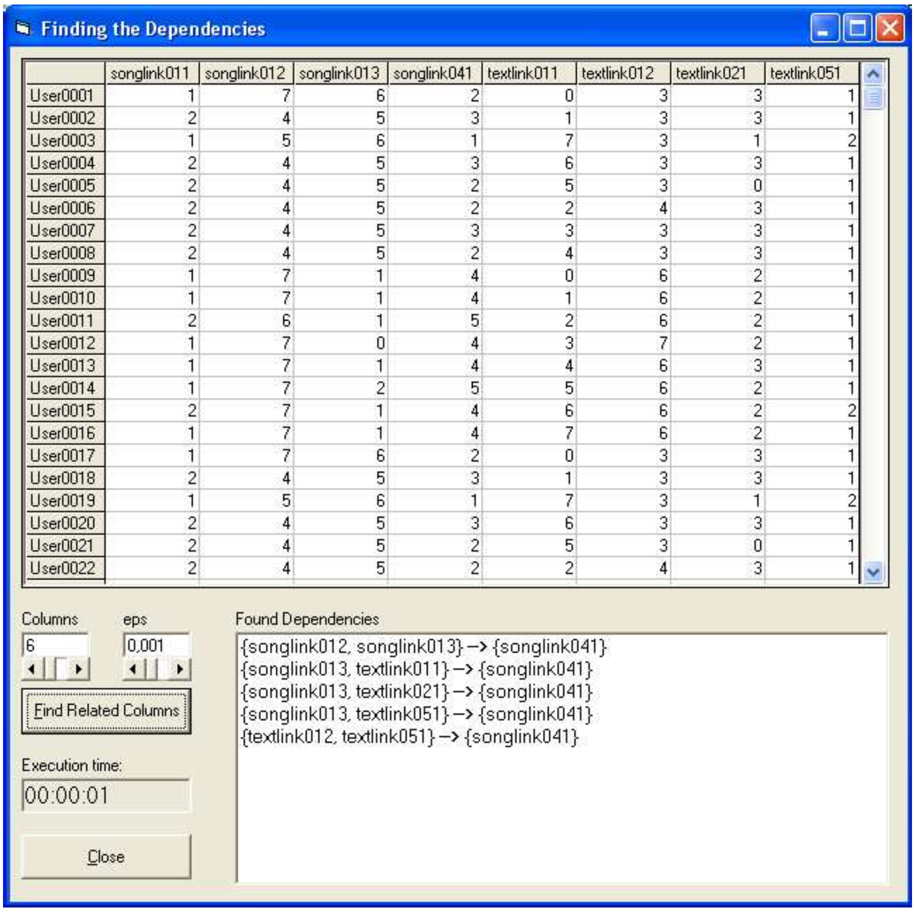
The represented realization of the proposed algorithm is applied to the data obtained from a WEB based client/server system that contains an archival fund with folklore materials of the Folklore Institute of BAS. The investigated archive keeps detailed information of the documents and materials, which can be downloaded by the users and contain audio, video and text information. The aim is to find the items, i.e. the materials which can be used to define the groups of users. We consider that the unknown original dataset M consists of the data that should be obtained if each user ranks the materials, which downloads and gives true rank. The dataset M^* consists of the real data.

For example, let the users can rank the materials with the integer values between 1 and 7. The value 0 means that the user is downloaded in, with the material but he does not give a rank. Some results are shown in Figure 1.

The described algorithm can be modified such that to discover only the dependencies $A \to b$ with attributes set A of minimal size for a given threshold ε. The results in figure 1 are obtained from the execution of this version of the algorithm. The application we realize gave possibility to determine the threshold value as well as the maximal number of the attributes in the left-hand side of the dependencies.

It is necessary to perform an additional processing for forming the groups of users. This process includes finding a substantial number of rows coinciding in the attributes of A, if $A \to b$ is valid.

The found dependencies are valid only in the present state of the database and, therefore, describe the content of the database precisely. They may become invalid, if the database changes. Therefore, we have to maintain the discovered knowledge, if we use it more than once. Since in the situation we consider the rows can be added in result of the appearance of new users, the dependencies may become invalid. Therefore, each dependence is checked if it is still valid. If not, then the dependence has to be replaced by a minimal dependence which is valid, if such dependence exists. In this case we can use



Finding the Dependencies

	songlink011	songlink012	songlink013	songlink041	textlink011	textlink012	textlink021	textlink051
User0001	1	7	6	2	0	3	3	1
User0002	2	4	5	3	1	3	3	1
User0003	1	5	6	1	7	3	1	2
User0004	2	4	5	3	6	3	3	1
User0005	2	4	5	2	5	3	0	1
User0006	2	4	5	2	2	4	3	1
User0007	2	4	5	3	3	3	3	1
User0008	2	4	5	2	4	3	3	1
User0009	1	7	1	4	0	6	2	1
User0010	1	7	1	4	1	6	2	1
User0011	2	6	1	5	2	6	2	1
User0012	1	7	0	4	3	7	2	1
User0013	1	7	1	4	4	6	3	1
User0014	1	7	2	5	5	6	2	1
User0015	2	7	1	4	6	6	2	2
User0016	1	7	1	4	7	6	2	1
User0017	1	7	6	2	0	3	3	1
User0018	2	4	5	3	1	3	3	1
User0019	1	5	6	1	7	3	1	2
User0020	2	4	5	3	6	3	3	1
User0021	2	4	5	2	5	3	0	1
User0022	2	4	5	2	2	4	3	1

Columns: 6

eps: 0,001

Execution time: 00:00:01

Found Dependencies

{songlink012, songlink013} –> {songlink041}
{songlink013, textlink011} –> {songlink041}
{songlink013, textlink021} –> {songlink041}
{songlink013, textlink051} –> {songlink041}
{textlink012, textlink051} –> {songlink041}

Fig. 1. Some results for exemplary data about 64 users

the algorithm similar of one represented in [3] but in the present version we are interested in the dependence which is determined completely in obtained dataset. It is described in Algorithm 2. We assume that the dataset M^* consists of the data after the addition the new rows in the dataset M.

Algorithm 2

Input: datasets M, M^* with the set of n attributes Ω; $A \to b$ is valid in M; $A \subset \Omega$, $A = \{a_1, \ldots, a_i\}$; $b \in \Omega$; threshold ε

Output: the enlargement A' of the set of attributes A such that $A' \to b$ holds in M^*

```
1) $A' = A$
2) $X = \Omega \backslash \{A \cup \{b\}\} = \{x_1, \ldots, x_p\}$
3) $p = |X| = n - i - 1$
4) **while** $p > 0$ and $e(A' \to b) > \varepsilon$
```

```
**for each** $x_k \in X$ **do**
$\delta_k = |\ \mathrm{F}(\pi_{a_1,\ldots,a_i}(M^*)) - \mathrm{F}(\pi_{a_1,\ldots,a_i,b}(M^*))|$
$\delta_d = \mathsf{min}_k(\delta_k)$ for $k = 1, \ldots, p$
$X = X \backslash \{x_d\}$
$A' = A' \cup \{x_d\}$
$p = p - 1$
**if** $e(A' \rightarrow b) \leq \varepsilon$ **then**
**output** $A' \rightarrow b$
```

The Algorithm 2 iterates until there are no more attributes to be added in the enlargement of the set A or until the searched dependence is found.

5. Analysis. We use the algorithm developed in [12] to compute the correlation fractal dimension D_2 of a given dataset which is an $O(m)$ algorithm, where m is the number of rows in the dataset. The computation of the information fractal dimension is based on a similar algorithm. Consequently, the algorithms described in this paper are linear on the number of rows m in the dataset, i.e. the number of points in the dataset in n-dimensional space.

The worst case time complexity of the execution of the Algorithm 1 for each $b \in \Omega$ with respect to the number of the attributes is exponential, but this is inevitable since the number of the minimal dependencies can be exponential in the number of attributes [9].

6. Comparison with other algorithms. The improved inference of FDs in [1] by using functional independencies leads to minimizing the number of accesses to the database during the discovery of FDs and their maintenance. However, this approach can not be directly applied to approximate dependencies.

The algorithm for discovering approximate dependencies proposed in [7] is linear on the number of tuples in the relation. It is based on partitioning the set of rows with respect to their attribute values and the definition of the approximate dependency is based on the minimum number of tuples that need to be removed from the relation for a given FD to hold in the relation. The time complexity for computing the partitions is $O(s.m)$, where s is the number of partitions and the time complexity for approximate validity testing is $O(m)$. The Algorithm 1 which is described in the present paper is not based on the partitions and tests the validity of the dependencies according to the fractal dimension of the corresponding datasets. These characteristics make the discovery of the dependencies fast for a large number of rows and suitable for situations in which the identification of the erroneous or exceptional rows is not necessary or possible.

The Algorithm 1 and the algorithm in [7] search for dependences in a breadth-first manner, i.e. they start the search from 1-element sets of attributes and continue with larger attribute sets. The small-to-large direction of the algorithms can be used to prune the search space efficiently. Additional pruning criteria requiring additional computation are applied in [7]. The computing of the partitions is avoided in Algorithm 1 and, therefore, a pruning procedure can be omitted.

7. Conclusion. We have presented an algorithm for finding the error-correcting functional dependencies in case of unknown original dependencies. The approach is based on considering approximate dependencies in the dataset obtained after the transmission.

The described way for defining the approximate dependency is to use the fractal dimension as an indicator of the existence of a correlation between the attributes in the dataset.

Currently, we are investigating the usage of data dependencies in collaborative filtering.

Acknowledgment. The authors wish to thank prof. G. O. H. Katona for the helpful discussions.

REFERENCES

[1] S. Bell. Discovery and Maintenance of Functional Dependencies by Independencies, In Proceedings of the First International Conference on Knowledge Discovery and Data Mining, 1995, 27–32.

[2] A. Belussi, C. Faloutsos. Estimating the Selectivity of Spatial Queries Using the 'Correlation' Fractal Dimension. In: Proceedings of the International Conference on Very Large Databases, 1995, 299–310.

[3] 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.

[4] J. Demetrovics, G. O. H. Katona, D. Miklós. Functional Dependencies Distorted by Errors, Discrete Mathematics (accepted).

[5] K. Falconer. Fractal Geometry: Mathematical Foundations and Applications, John Wiley & Sons, Chichester, 1990.

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

[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] B. Mandelbrot. The Fractal Objects: Form, Fortuity and Dimension, University of Sofia "St. Kliment Ohridski", Sofia, 1996 (in Bulgarian).

[9] H. Mannila, K. J. Raiha. On the Complexity of Inferring Functional Dependencies. *Discrete Applied Mathematics*, **40** (1992), 237–243.

[10] J. Peneva. Databases, Regalia 6, 2004 (in Bulgarian).

[11] D. M. Pennock, E. Horvitz, C. L. Giles. Social Choice Theory and Recommender Systems: Analysis of the Axiomatic Foundations of Collaborative Filtering. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence, 2000, 729–734.

[12] C. Traina, A. Traina, L. Wu, C. Faloutsos. Fast Feature Selection Using the Fractal Dimension. In: XV Brazilian Symposium on Databases, 2000.

Galina Bogdanova
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
5000 Veliko Tarnovo, P.O.Box: 323
e-mail: galina@moi.math.bas.bg

Tsvetanka Georgieva
University of Veliko Tarnovo
Department of Information Technologies
e-mail: cv.georgieva@uni-vt.bg

ОТКРИВАНЕ НА ФУНКЦИОНАЛНИТЕ ЗАВИСИМОСТИ, ПОПРАВЯЩИ ГРЕШКИ ПРИ НЕИЗВЕСТНИ ПЪРВОНАЧАЛНИ ЗАВИСИМОСТИ

Галина Т. Богданова, Цветанка Л. Георгиева

Откриването на връзки между дадена функционална зависимост в набор от данни и съответната функционална зависимост в набора от данни, който е получен след предаването му по канал с шум, е важна задача за анализирането на една база данни. В случая на разработване на данни (data mining) само получените данни са известни и целта е да се направят изводи за функционалните зависимости на напълно неизвестния първоначален набор от данни. В настоящата статия е представен алгоритъм за намиране на функционалните зависимости, поправящи грешки в този случай. За решението на проблема се използват приблизителни зависимости, чиято дефиниция е основана на фракталната размерност на съответните набори от данни.