Serdica J. Computing **3** (2009), 319–334

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

METHODS FOR INVESTIGATION OF DEPENDENCIES BETWEEN ATTRIBUTES IN DATABASES*

Tsvetanka Georgieva

Abstract. This paper surveys research in the field of data mining, which is related to discovering the dependencies between attributes in databases. We consider a number of approaches to finding the distribution intervals of association rules, to discovering branching dependencies between a given set of attributes and a given attribute in a database relation, to finding fractional dependencies between a given set of attributes and a given attribute in a database relation, and to collaborative filtering.

1. Introduction. Data mining is a scientific domain that emerged due to the need to analyse the data accumulated in the course of the daily activities of a particular organization and saved in databases in order to find some characteristic patterns, rules, and hidden correlation relationships between different attributes which, in its turn, would support decision making.

ACM Computing Classification System (1998): H.2.8.
Key words: Data mining, association rules, dependency discovery.

*This article presents the principal results of the doctoral thesis "Methods for investigation of dependencies between attributes in databases" by Tsvetanka Georgieva (St. Cyril and St. Methodius University of Veliko Tarnovo), successfully defended before the Specialised Academic Council for Informatics and Mathematical Modelling on 23 February, 2009.

The methods for data mining are successfully utilized for discovering general characteristics in order to receive information from a higher level (for example rules, dependencies, etc.), needed for making decisions or investigating, predicting or modeling the dependencies that generated the data. Therefore, developing methods for analysing and extracting dependencies from databases is a crucial task.

This paper aims at surveying existing research in the field of data mining, which is related to discovering dependencies between attributes in databases. We examine various approaches to finding the distribution intervals of association rules, to discovering branching dependencies between a given set of attributes and a given attribute in a database relation, to finding fractional dependencies between a given set of attributes and a given attribute in a database relation, and to collaborative filtering.

The rest of the paper is organized as follows: Section 2 reviews some basic concepts related to data warehousing and data mining, as well as existing solutions to the problem of discovering association rules and their distribution in time; Section 3 dwells on the properties of branching and fractional dependencies; Section 4 is devoted to methods for collaborative filtering.

2. Discovering distribution intervals of association rules in time. A data warehouse [28], [43] is defined as a subject-oriented, integrated, time-variant, non-volatile collection of data, supporting decision making. Typically the data warehouse is maintained separately from the operational database of an organization. Data warehouses support on-line analytical processing (OLAP) whose functional and performance requirements are quite different from those of the on-line transaction processing (OLTP) applications that are traditionally supported by operational databases.

Data mining [5], [10], [20], [22] is the process of extracting interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from large information repositories such as: relational databases, data warehouses, XML repositories, etc. Furthermore, data mining is known as one of the core processes of Knowledge Discovery in Database (KDD).

Association rule mining is a form of data mining aimed at discovering interesting correlation relationships among attributes of the data under analysis. The rules thus discovered may facilitate decision making in a number of areas. An association rule shows the recurrent patterns of a given set of data items in

the database. Association rules are introduced in [2] and they are employed for determining the relationships between a set of items in a database. These relationships reveal properties of the data which are based on the frequent occurrence of the data items.

OLAP-based association rules mining integrates OLAP technology and association rules mining which facilitates mining of interesting knowledge in data cubes because data mining can be performed in the multidimensional and multilevel abstraction space in a data cube [25], [29].

[6], [14] suggest an application utilizing OLAP operations to analyse the data in a Web-based client/server system that contains an archive of folklore materials. This archive stores detailed information about documents and materials, which can be downloaded by users, and contains audio, video and text information.

In [7], [14] an application that discovers association rules by using a data cube structure and applying OLAP operations is proposed. The application thus developed enables us to perform association analysis of daily downloads of folklore materials according to the dimensions of interest. Discovered association rules can then be displayed in different ways when the user needs to view and analyse the rules from different aspects.

In addition to association rules, their distribution in time has important practical applicability, since a rule provides no information about the distribution of the items that generated its support. [3] describes an algorithm which finds the distribution intervals of association rules in time and computes the fractal dimension and a significant change of its value indicates the beginning of a new interval. The previous research on the fractal mining of distribution intervals of association rules uses a relation table-based structure and requires a multiple scan of the data.

One of the major advantages of OLAP mining is using data extracted from data warehouses. The data is loaded into a data warehouse after it has previously been integrated, consolidated, cleaned, and transformed. This has fostered a study on the mining of distribution intervals of association rules in time using the data cube structure and applying the OLAP operations. The approach to finding separate intervals proposed in [12], [14] explores the data cube structure. A computer program implementation of the algorithm for discovering the distribution interval of the association rules in time is demonstrated.

3. Branching and fractional dependencies in databases. The discovery of dependencies is an important activity in the area of data mining and is used in many applications for knowledge discovery in databases. Functional dependencies are relationships between attributes of a database relation. Some functional dependencies are defined while the database is being designed and are used to reduce the amount of the redundant data. Then constraints are settled to support the referential integrity. There are few constraints, however, and often they are too general since they have to be 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 [4], [11], [15], [27], [30], [34], [44]. The main motivation for discovery of the functional dependencies, which hold in the current instance of a relation, is to discover valuable knowledge about the structure of the relation instance, and thus to support the various experts (e.g. managers, analysts, etc.) in making well-informed decisions faster.

In some cases, a given functional dependency is not valid for a small number of tuples. Such a functional dependency can be thought of as approximate, i.e. it almost holds. One way of defining the approximate dependency is based on the minimal number of tuples that need to be removed from the relation for the relevant functional dependency to hold [31].

Approximate functional dependencies also provide valuable knowledge of the structure of the current instance of the relation. The discovery of such knowledge can be valuable for analysing the data that is contained in the database by domain specialists. Functional and approximate dependencies can also be applied in the area of query optimization [23], [35] and reverse engineering [1], [24].

A functional dependency requires the values in a given set of attributes to uniquely determine the value of a given attribute. In [18] a branching dependency, which is a more general dependency than the functional dependency, is introduced. It is not possible to use branching dependencies to decrease the size of the stored relation, as in the case of functional dependencies, but it is possible to constrain the range of the values of the attributes.

Some theorems valid for functional dependencies are generalized to apply to branching dependencies in [18]. Moreover, some implications among branching dependencies are investigated. In [19] the sets of attributes that (p, q)-depend on sets A of attributes are considered and estimates are made of the minimal number of tuples in a relation for which these sets of attributes are obtained ($1 \leq p \leq q$, integers).

In [13], [14] a modified form of the branching dependency defined in [18]

is considered. The modified branching dependency enables us to determine the maximal number of different values of a given attribute b corresponding to one or more different values of a given set of attributes A in the relation. The motivation for addressing the problem of finding these dependencies stems from their applications in data mining, which aims at discovering interesting and useful patterns in large databases to support future decision making.

As a motivating example, we consider a database of products and the customers who purchased them or Web pages and users who visited them. Discovering that an arbitrary p number of products have attracted a total of at most q new users over a given period of time can be crucial for supporting the decision-making process.

We obtain a fractional branching dependency by adding the requirement for b to functionally determine A. In the case of the fractional dependency, we can determine the maximal number of different values of$b$ corresponding to p different values of the attributes of A, but we consider only these values of b that remain after the elimination of the values of b which result in the maximum for $p-1$. This knowledge is additional information that may be useful to domain specialists when they analyse the current content of a database.

For example, it would be useful to discover that the pth new product offered (service, promotion) has attracted at most c_p new users who have not been attracted by some of the previous products, on condition that each following product has been selected so as to attract a maximal number of new customers.

Other reasons for discovering the branching and fractional dependencies include maintaining detailed information in store for optimization of some queries or extracting information for decreasing the range of the values of the attributes for database reverse engineering.

In [13], [14] 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 in a database relation is proposed. In addition, a fractional dependency and a fractional branching dependency are defined and some properties of these dependencies are examined and proven. An algorithm for finding all fractional dependencies between a given set of attributes and a given attribute in a database relation is suggested. A general case of an arbitrary relation is examined, and a case study of a particular relation is also presented, where the task of discovering fractional dependencies is significantly simplified.

In [13], [14] the task of finding all branching dependencies between a

given set of attributes A and a given attribute b is considered. For this purpose a minimum branching dependency is defined so that the validity of all the branching dependencies between A and b can be established if all minimal dependencies are known. Moreover, some properties of the branching dependencies are proven which enables us to prune some values of p and q during the search for the branching dependencies between A and b and to create an efficient algorithm.

Some properties of the fractional branching dependencies and the fractional dependencies are examined. Building upon these properties, an algorithm is proposed which is designed to find the fractional dependencies in the general case of an arbitrary relation. Some methods for solving this problem are proposed in the case of data constraints on the attributes analysed.

In [13], [14] the implementations of the algorithms for finding the minimal branching dependencies and fractional dependencies in databases are presented.

4. Collaborative filtering. The increasing number of various products and services offered by e-commerce Web sites requires the implementation of recommender systems. These systems provide customers with the necessary information to facilitate their choice of products. The large number of visitors to Web sites and the amount of data accumulated about them provide ample opportunities to organize and recommend information. Various technologies have been developed to help customers to quickly find the most appropriate products.

Collaborative filtering techniques [9], [26,] [33], [37], [39], [40] apply knowledge discovery algorithms to make personalized recommendations for information to users. For this purpose, the system maintains a database which stores users' ratings of items. The major challenges collaborative filtering faces are the accuracy of the recommendations to users and the scalability of the algorithms used to compute the predicted user ratings of items. The quality (or accuracy) of the recommendations can be influenced by a variety of factors, such as sparsity and noise due to the fact that very few people rate exactly the same items or that people may not give ratings or may not give true ratings. Collaborative filtering algorithms require the computation of recommendations with a growing number of users and items, which results in scalability problems.

Many collaborative filtering algorithms have been proposed by researchers. They can be divided into two general classes – memory-based and model-based algorithms. Memory-based algorithms operate over the entire database of user ratings of the items to generate predictions [9], [26], [37]. Typically, these sys-

tems calculate a similarity measure between the active user and the other users, and then predictions are generated by using a weighted aggregate of user ratings. The most popular memory-based algorithm used in collaborative filtering is the Pearson algorithm proposed in [37]. Improvement of the accuracy of the recommendations can be achieved by applying Case amplification [9] which transforms the Pearson correlation coefficients. The main challenges to memory-based collaborative filtering are scalability and data sparsity. Usually the similarity between users is based on the overlap of the users' ratings and thus data sparsity influences the reliability of the computed similarity. In addition, when the database becomes larger, the online computation of the similarity measure cannot be performed efficiently.

Model-based collaborative filtering algorithms [33], [39], [40] use a database that stores user ratings, to develop a model, which is then consulted for predictions. Some of these systems are based on Bayesian networks, clustering [9], and rule-based approaches [39]. As with the item-based algorithms proposed in [40], it is the similarity between items which is considered first, and then the predictions are obtained as weighted averages of the user ratings. The Slope One algorithms [33] pre-compute the average difference between the ratings of paired items rated by the same users. In comparison to memory-based schemes, model-based algorithms require more time to build a model but can retrieve the query result at the prediction generation step faster.

Recommendation technologies applied in different areas and examples of recommendation systems used in e-commerce are thoroughly reviewed in [21], [41], [42].

In [8], [14] the issues of collaborative filtering are examined by applying a method for discovering error-correcting functional dependencies using the fractal dimension. Error-correcting functional dependencies are introduced in [16] as functional dependencies which are valid in spite of the errors in the data after the transmission. In [17] the authors consider the case in which only the received data is known, and the aim is to make inferences about the functional dependencies of the completely unknown original dataset.

The motivation for using the fractal dimension is to decrease the stored information needed to generate predictions, in order to increase the efficiency of the online computations and to keep the accuracy. [8], [14] examine a fractal-based algorithm for discovering the error-correcting functional dependencies and their application in collaborative filtering. A vector quantization is used, which is defined so as to satisfy the Nearest Neighbor Condition for the Hamming dis-

tance and the Centroid Condition for the squared-error measure. Hence, elements which coincide in most of the dataset's attributes are replaced by a representative point averaging their values. The relation between the performance of the vector quantization and the fractal dimension of a dataset is used [32] in order to make conclusions about the dataset which we have to choose.

The recommendations to users are computed by discovering items whose ratings determine other items to the greatest extent. The interrelations between the items are determined by computing the fractal dimension of datasets retrieved from the database of user ratings for these items.

Results from experiments conducted to assess the accuracy and scalability of the proposed fractal-based algorithm for collaborative filtering are presented. The proposed method is based on computing the fractal dimension of a dataset retrieved from the database of user ratings for the items. Since redundant data storage has to be avoided, prediction generation query retrieval is performed much faster by means of the fractal-based method.

4.1. Accuracy evaluating measures. The most common measure used to evaluate the accuracy of collaborative filtering is the *Mean Absolute Error*. The Mean Absolute Error measures the average absolute deviation between a predicted rating and the user's real rating. This measure is defined in the following way:

$$MAE_u = \frac{1}{|P_u|} \sum_{b \in P_u} |p_{u,b} - r_{u,b}|,$$

where $|P_u|$ is the number of computed ratings in the test dataset for user u; $p_{u,b}$ is the predicted rating of item b given by user u; $r_{u,b}$ is the real rating given by user u for item b; P_u is the set of items for which the ratings of user u are computed.

We apply the *All But One Mean Average Error* (MAE), which results from the consecutive exclusion of one rating from the dataset and every time the hidden rating is predicted.

Besides the measure described above, the *Root Mean Squared Error* (RMSE) is used in our experiments. It is defined as:

$$RMSE_u = \sqrt{\frac{1}{|P_u|} \sum_{b \in P_u} (p_{u,b} - r_{u,b})^2}.$$

The result indicates more clearly the existence of larger deviations (errors).

4.2. Methods. *Global Average* is the average of all ratings in the dataset. *User Average* is used to compute the average value of the ratings provided by a given user u:

$$p_u = \frac{1}{|M_u|} \sum_{b \in M_u} r_{u,b},$$

where M_u is the set of items rated by user u.

Item Average computes the rating of a given item b as the average of all ratings for this item:

$$p_b = \frac{1}{|M_b|} \sum_{u \in M_b} r_{u,b},$$

where M_b is the set of users who rated item b.

When employing the *Double Average* scheme to predict the rating of a given user for a given item, the average rating for the item is computed, as well as the average rating of the respective user, and then the two values are averaged out:

$$p_{u,b} = \frac{p_u + p_b}{2}.$$

As to the *Bias From Mean* [26], the prediction is based on the average of the user ratings and the average deviation of the user mean for the given item:

$$p_{u,b} = p_u + \frac{1}{|M_b|} \sum_{v \in M_b} (r_{v,b} - p_v).$$

The memory-based *Pearson algorithm* [37] computes the prediction about the rating of user u for item b by computing the weighted sum of the ratings of the rest of the users:

$$p_{u,b} = p_u + \frac{\sum_{v \in M_b} w(u,v)(r_{v,b} - p_v)}{\sum_{v \in M_b} |w(u,v)|},$$

where $w(u,\ v)$ is the similarity measure, defined on the basis of the Pearson's correlation:

$$corr_{u,v} = \frac{\sum_{a \in M_u \cap M_v} (r_{u,a} - p_u)(r_{v,a} - p_v)}{\sqrt{\sum_{a \in M_u \cap M_v} (r_{u,a} - p_u)^2 \sum_{a \in M_u \cap M_v} (r_{v,a} - p_v)^2}}.$$

We employ Case amplification [9], where $w(u,v) = corr_{u,v}\ |corr_{u,v}|^{\rho-1}$ with $\rho = 2.5$.

Comparatively, we use the *item-based approach* suggested in [40], which is based on the following adjusted cosine item-based measure:

$$sim_{b,a} = \frac{\sum\limits_{u \in M_{b,a}} (r_{u,b} - p_u)(r_{u,a} - p_u)}{\sum\limits_{u \in M_{b,a}} (r_{u,b} - p_u)^2 \sum\limits_{u \in M_{b,a}} (r_{u,a} - p_u)^2},$$

where $M_{b,a}$ is the set of users who rated items b and a.

The prediction is obtained as a weighted sum of the respective similarity measures:

$$p_{u,b} = \frac{\sum\limits_{a \in M_u} sim_{b,a} r_{u,a}}{\sum\limits_{a \in M_u} |sim_{b,a}|}.$$

The *Weighted Slope One* algorithm [33] computes the average deviation of item b in relation to item a:

$$dev_{a,b} = \sum_{u \in M_{a,b}} \frac{r_{u,a} - r_{u,b}}{|M_{a,b}|}.$$

The prediction is computed as follows:

$$p_{u,b} = \frac{\sum\limits_{a \in M_u \setminus \{b\}} (dev_{a,b} + r_{u,a})|M_{a,b}|}{\sum\limits_{a \in M_u \setminus \{b\}} |M_{a,b}|}.$$

We present the results from the accuracy evaluation (Table 1), as well as the results from the scalability evaluation (Figure 1) of the fractal-based method for collaborative filtering.

Experimental results demonstrate that this method improves the accuracy (Table 1) and performance (Figure 1) of the filtering as compared to other memory-based and model-based methods for collaborative filtering. The methods used for comparison have become popular due to a reasonable compromise between accuracy, performance, and the simplicity of their implementation.

Algorithm	FolkloreDB		MovieLens		Netflix	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Global Average	1.914784	2.127767	0.944700	1.125668	0.938515	1.127432
User Average	1.864951	2.095598	0.826226	1.030811	0.914297	1.092452
Item Average	1.463987	1.712658	0.798958	1.000070	0.830840	1.036403
Double Average	1.586926	1.796998	0.782583	0.966809	0.790203	1.003288
Bias From Mean	1.402642	1.671725	0.769381	0.975374	0.781415	1.038440
Pearson	0.979386	1.262602	0.765675	1.128491	0.767586	1.199988
Adjusted Cosine Item-based	1.074036	1.449975	0.759785	0.945644	0.754883	1.192057
Weighted Slope One	1.475316	1.760152	0.731040	0.926571	0.732227	1.213624
Fractal-based	0.701821	1.174057	0.710712	0.924482	0.712909	1.008744

Table 1. All But One Mean Average Error and Root Mean Squared Error for the FolkloreDB [8], MovieLens [38], Netflix [36] datasets

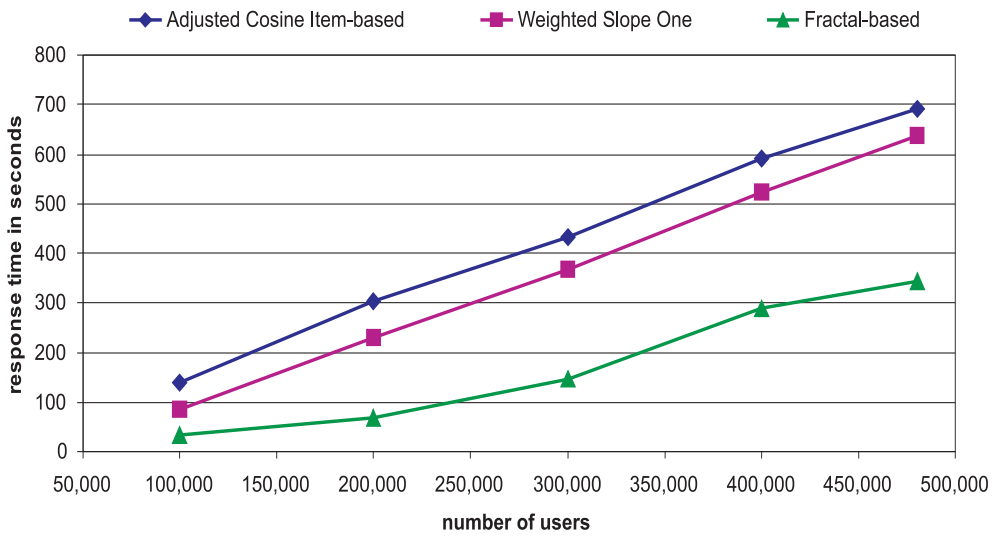

Fig. 1. Recommendation time for the Netflix dataset

5. Conclusion. The growing importance of the issues discussed in this paper is triggered by the contemporary trends in designing highly effective automated systems for processing databases for the purpose of discovering dependencies in them.

The process of data mining finds application in different areas such as Internet technologies, the insurance business, telecommunications, various industries, healthcare, etc.

REFERENCES

[1] Abbasifard M. R., M. Rahgozar, A. Bayati, P. Pournemati. Using Automated Database Reverse Engineering for Database Integration. In: Proceedings of World Academy of Science, Engineering and Technology, 2006, 13–17.

[2] Agrawal R. , T. Imielinski, A. Swami. Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, 1993, 207–216.

[3] Barbara D. , Z. Nazeri. Fractal Mining of Association Rules over Interval Data. Technical Report, George Mason University, 2000, 9.

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

[5] Berkhin P. Survey of Clustering Data Mining Techniques. `http://citeseer.ist.psu.edu/berkhin02survey.html`, 2002.

[6] Bogdanova G., Tsv. Georgieva. Analysing the Data in OLAP Data Cubes. *International Journal on Information Theories and Applications*, **12** (2005), No 4, 335–342.

[7] Bogdanova G. , Tsv. Georgieva. An Application for Discovering the Association Rules in OLAP Data Cubes, *Automatica and Informatics*, **40** (2006), No 4, 29–33.

[8] Bogdanova G., Tsv. Georgieva. Using Error-correcting Dependencies for Collaborative Filtering, *Data and Knowledge Engineering*, Elsevier, **66** (2008), No 3, 402–413,
http://dx.doi.org/10.1016/j.datak.2008.04.008.

[9] Breese J. , D. Heckerman, C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, 43–52.

[10] Chen M.-S., J. Han, P. S. Yu. Data Mining: An Overview from a Database Perspective. *IEEE Transactions on Knowledge and Data Engineering*, **8** (1996), 866–883.

[11] Coen G. Database Lexicography. *Data and Knowledge Engineering*, Elsevier, **42** (2002), No 3, 293–314.

[12] Georgieva Tsv. Algorithm for Discovering the Distribution Intervals of the Association Rules in OLAP Data Cubes. *Mathematica Balkanica*, **20** (2006), No 3–4, 387–397.

[13] Georgieva Tsv. Discovering Branching and Fractional Dependencies in Databases. *Data and Knowledge Engineering*, Elsevier, **66** (2008), No 2, 311-325, http://dx.doi.org/10.1016/j.datak.2008.04.002.

[14] Georgieva Tsv. Methods for investigation of dependencies between attributes in databases, Ph.D. Thesis, University of Veliko Tarnovo, 2009, 128.

[15] Flach P. A. , I. Savnik. Database Dependency Discovery: A Machine Learning Approach. *AI Communications*, **12** (1999), No 3, 139–160.

[16] Demetrovics J. , G.O.H. Katona, D. Miklós. Functional Dependencies in Presence of Errors. In: Foundations of Information and Knowledge Systems (FoIKS 2002), Lecture Notes in Computer Science, **2284**, Springer, 2002, 85–92.

[17] Demetrovics J., G.O.H. Katona, D. Miklós. Functional Dependencies Distorted by Errors. *Discrete Applied Mathematics*, **156** (2008), No 6, 862–869.

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

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

[20] Fayyad U. , G. Piatetsky-Shapiro, P. Smyth. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 1996, 37–54.

[21] Felfernig A., S. Gordea, D. Jannach, E. Teppan, M. Zanker. A Short Survey of Recommendation Technologies in Travel and Tourism. *ÖGAI Journal*, **25** (2006), No 2, 1–7.

[22] Frawley W., G. Piatetsky-Shapiro, C. Matheus. Knowledge Discovery in Databases: An Overview. *AI Magazine*, 1992, 213–228.

[23] Giannella C., M. Dalkilic, D. P. Groth, E. L. Robertson. Improving Query Evaluation with Approximate Functional Dependency Based Decompositions. In: Proceedings of the 19th British National Conference on Databases: Advances in Databases, 2002, 26–41.

[24] Hainaut J.-L. Introduction to Database Reverse Engineering. LIBD – Laboratory of Database Application Engineering, Institut d'Informatique, University of Namur, 2002.

[25] Han J. Towards On-Line Analytical Mining in Large Databases. SIGMOD Record (ACM Special Interest Group on Management of Data), 1998, 97–108.

[26] Herlocker J., J. Konstan, A. Borchers, J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In: Proceedings of Research and Development in Information Retrieval, 1999, 230–237.

[27] Huhtala Y., 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.

[28] Inmon W. H. Tech Topic: What is a Data Warehouse? Prism Solutions, Inc. 1995.

[29] Kamber M., J. Han, J. Chiang. Using Data Cubes for Metarule-Guided Mining of Multi-Dimensional Association Rules. Technical Report, CMPT–TR–97–10, School of Computing Sciences, Simon Fraser University, 1997, 6 pp.

[30] Kantola M., H. Mannila, K.-J. Raiha, H. Sirtola. Discovering functional and inclusion dependencies in relational databases. *International Journal of Intelligent Systems*, 1992, 591–607.

[31] Kivinen J., H. Mannila. Approximate Inference of Functional Dependencies from Relations. *Theoretical Computer Science*, **149** (1995), 129–149.

[32] Kumaraswamy K., V. Megalooikonomou, C. Faloutsos. Fractal Dimension and Vector Quantization. *Information Processing Letters*, **91** (2004), No 3, 107–113.

[33] Lemire D., A. Maclachlan. Slope One Predictors for Online Rating-Based Collaborative Filtering. In: Proceedings of the SIAM International Conference on Data Mining, 2005, 21–23.

[34] Matos V., B. Grasser. SQL-based Discovery of Exact and Approximate Functional Dependencies. *ACM SIGCSE Bulletin*, **36** (2004), No 4, 58–63.

[35] Nambiar U., S. Kambhampati. Mining Approximate Functional Dependencies and Concept Similarities to Answer Imprecise Queries. In: Proceedings of the Seventh International Workshop on the Web and Databases, 2004, 73–78.

[36] Netflix Prize Website, `http://www.netflixprize.com`

[37] Resnick P., N. Iacovou, M. Suchak, P. Bergstorm, J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: Proceedings of the Conference on Computer Supported Cooperative Work, 1994, 175–186.

[38] Riedl J., J. Konstan. GroupLens Research Project. MovieLens dataset, `http://www.grouplens.org/data`

[39] Sarwar B., G. Karypis, J. Konstan, J. Riedl. Analysis of Recommendation Algorithms for E-Commerce. In: Proceedings of the Second ACM conference on Electronic commerce, 2000, 158–167.

[40] Sarwar B., G. Karypis, J. Konstan, J. Riedl. Item-based Collaborative Filtering Recommendation Algorithms. In: Proceedings of the 10th International Conference on World Wide Web, 2001, 285–295.

[41] Schafer J., J. Konstan, J. Riedl. E-Commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, **5** (2001), No 1–2, 115–153.

[42] Wei K., J. Huang, S. Fu. A Survey of E-Commerce Recommender Systems. In: Proceedings of the International Conference on Service Systems and Service Management, 2007, 1–5.

[43] Widom J. Research Problems in Data Warehousing. In: Proceedings of 4th International Conference on Information and Knowledge Management (CIKM), 1995, 25–30.

[44] Wyss C., C. Giannella, E. Robertson. FastFDs: A Heuristic-Driven, Depth-First Algorithm for Mining Functional Dependencies from Relation Instances. In: Proceedings of Third International Conference of the Data Warehousing and Knowledge Discovery, 2001, 101–110.

Department of Information Technologies
St. Cyril and St. Methodius University of Veliko Tarnovo
3, Architect Georgi Kozarov Str.
Veliko Tarnovo, Bulgaria
e-mail: cv.georgieva@uni-vt.bg

Received May 28, 2009
Final Accepted June 30, 2009