Bayes Net Analysis to Support Database Design and Normalization

Nittaya Kerdprasop
Data Engineering Research Unit,
School of Computer Engineering,
Suranaree University of Technology, Thailand
nittaya@sut.ac.th

Kittisak Kerdprasop
Data Engineering Research Unit,
School of Computer Engineering,
Suranaree University of Technology, Thailand
kerdpras@sut.ac.th

***Abstract*—Knowledge discovery, also known as data mining, has gained much interest from diverse sectors due to their great potential on revealing potentially useful relationships. Among many possible applications, we focus our research on the database design and analysis application. Functional dependency plays a key role in database normalization, which is a systematic process of verifying database design to ensure the nonexistence of undesirable characteristics. Bad design could incur insertion, update, and deletion anomalies that are the major cause of database inconsistency. In this paper, we propose a novel technique to discover functional dependencies from the database table. The discovered dependencies help the database designers covering up inefficiencies inherent in their design. Our discovery technique is based on the structure analysis of Bayesian network. Most data mining techniques applied to the problem of functional dependency discovery are rule learning and association mining. Our work is a novel attempt of applying the Bayesian network to this area of application. The proposed technique can reveal functional dependencies within a reduced search space. Therefore, computational complexity is acceptable.**

Keywords-functional dependency; knowledge discovery; data mining; Bayesian network; database design; normalization

I. INTRODUCTION

Database design methodology normally starts with the first step of conceptual schema design in which users' requirements are modeled as the entity-relationship (ER) diagram. The next step of logical design focuses on the translation of conceptual schemas into relations or database tables. The later step of physical design concerns the performance issues such as data types, indexing option, and other parameters related to the database management system. Conceptual schema and logical designs are two important steps regarding correctness and integrity of the database model. Database designers have to be aware of specifying thoroughly primary keys of tables and also determining extensively relationships between tables. Data normalization is a common mechanism employed to support database designers to ensure the correctness of their design.

Normalization transforms unstructured relation into separate relations, called normalized ones [9]. The main purpose of this separation is to eliminate redundant data and reduce data anomaly (i.e., data inconsistency as a result of insert, update, and delete operations). There are many different levels of normalization depending on the purpose of database designer. Most database applications are designed to be either in the third, or the Boyce-Codd normal forms in which their dependency relations [3] are sufficient for most organizational requirements**.**

The main condition to transform from one normal form to the next level is the dependency relationship, which is a constraint between two sets of attributes in a relation. Functional dependency constrains the determination uniqueness from one set of attribute values to the others.

Experienced database designers are able to elicit this kind of dependency information. But in some applications in which the business process and operational requirements are complex, this task of dependency analysis is tough even for the experienced ones. We thus propose to use the data mining technique called Bayesian network learning or Bayes net to help analyzing the database structure and then report the underlying functional dependencies. This work can also support the automatic induction of functional dependencies from legacy databases that design documents are no longer available. The rest of this paper is organized as follows. We discuss related work regarding the functional dependency discovery problem in Section 2. Then propose our methodology in Section 3. Running examples and experimentation appear in Sections 4 and 5, respectively. Finally, the last section concludes this paper with the mention of our future work.

II. RELATED WORK

The main objective of our study is the induction of functional dependency relationships from the database instances. It has long been the problem of interest among database researchers [16], [22], [23], [26], [29], [32], [33] because such relationships are abstract in their nature and hence can be easily missed out in the database design.

Silva and Melkanoff [29] was the first team attempting to discover functional dependencies (FDs) through the data mining technique. The complexity of discovering FDs from existing database instances has been studied by Mannila and Raiha [22], [23]. Early work on FD discovery handled the complexity problem by means of partitioning the set of rows according to their attribute values and perform a level-wise search for desired solution [15], [17], [26], [33]. The later work of Wyss et al. [33] and Atoum et al. [4] applied the minimal cover concept on equivalent classes.

Researchers in the application area of database reverse engineering are also interested in the same objective. Lee and

 ISBN: 978-1-61208-269-1

Yoo [20] proposed a method to derive a conceptual model from object-oriented databases. The final products of their method are the object model and the scenario diagram describing a sequence of operations. The work of Perez et al. [28] emphasized on relational object-oriented conceptual schema extraction. Their technique is based on a formal method of term rewriting. Rules obtained from term rewriting are then generated to represent the correspondences between relational and object-oriented elements. Researchers that focus their study on a particular issue of semantic understanding including Lammari et al. [19] who proposed a method to discover inter-relational constraints and inheritances embedded in a relational database. Chen et al. [7] also based their study on entity-relationship model. They proposed to apply association rule mining to discover new concepts leading to a proper design of relational database schema. They employed the concept of fuzziness to deal with uncertainty inherited with the association mining process. The work of Pannurat et al. [27] and Alashqur [1] are also in the line of association mining technique application to the database design.

Besides functional dependencies, other kinds of database relationships are also explored. De Marchi et al. [11] studied the problem of inclusion dependencies. Fan et al. [12] proposed the idea to capture conditional FDs. Calders et al. [6] introduced a notion of roll-up dependency to be applied to the OLAP context. Approximate FDs concept has been recently applied to different subfield of data mining such as decision tree building [18], data redundancy detection [2], and data cleaning [8], [24].

Our work is different from those in the literature in that our method of discovering FDs is based on the analysis of Bayes net structure [13], [14], [21], [30]. The work of Mayfield et al. [24] also consider applying Bayesian network but for a different purpose of correcting missing information. They perform stochastic process to simulate and estimate missing values. Our work does not require simulation; we instead base our discovery task solely on the existing database instances. Therefore, from the literature review we can state that our work is original in this area of problem.

III. PROPOSED METHODOLOGY

Functional dependencies between attributes of a relation express a constraint between two sets of attributes. For instance, the constraint $X \rightarrow Y$ states that the values of attributes in a set Y are fully determined by values of attributes in a set X. The obvious example is given the social security number, there is at most one individual associated with that number.

Discovering such constraints from the database instances requires extensive search over each pair of attribute values. We propose a methodology of employing Bayesian networks learning [5], [25] at this step. Bayesian network or Bayes net is a graphical model representing correlations among variables in the network structure. Relation attributes correspond to variables, which appear as nodes in the Bayes net. A Bayes net is a directed acyclic graph whose edges represent statistical dependencies.

Input: a set of database instances
Output: functional dependency rules
Steps:

/* *Learning Phase* */

1. Apply Bayesian learning algorithm to the set of database instances to form a network structure
2. Identify conditional independency with ICS search algorithm [26] and assign direction to the edges
3. Output causal Bayes net with additional conditional probability table associated with each node

/* *FDs Detection Phase* */

4. For an effect node (E) linking from a single causal node (C) or at most two causal nodes
5. Check conditional probability table of the effect node
6. If for each value of E there exists distinct value of C related to E with probability not less than 0.5 (regarding to existing values in the database instances),
   Then add a rule $C \rightarrow E$ to the FD rule set
7. If C contains two nodes (C1 and C2),
   Then repeat step 6 with $C1 \rightarrow E$ and $C2 \rightarrow E$
8. For an effect node (E) linking from multiple causal nodes (Cs)
9. Examine each edge linking from each C in Cs
10. If the criterion in steps 5-6 is satisfied by each and every edge,
    Then add a rule $Cs \rightarrow E$ to the FD rule set
11. Return the FD rule set as the output

Figure 1. Algorithm BayesFD for discovering functional dependencies from the database instances

With the proper search procedure, such as the ICS algorithm [31], and the use of conditional dependencies associated with each node, we can turn dependence relations between variables into causal relationship among nodes in the Bayes net.

We also apply heuristics on the consideration over each conditional dependency table associated with the child node to select proper functional dependencies from the Bayes net. For a strong dependency requirement, the minimum conditional probability can be set greater than 0.5. Our algorithm of FDs discovery, named BayesFD, is presented in Fig. 1.

IV. RUNNING EXAMPLES AND ANALYSIS RESULTS

We demonstrate the mechanism of our proposed method via the two examples.

Example 1. The database instances are given as shown in Table I.

TABLE I. EXAMPLE DATABASE

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b3	c2	d3
a3	b3	c2	d4

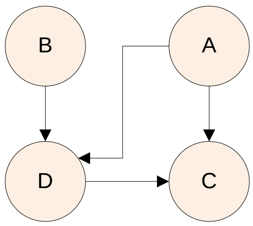

Figure 2. Bayesian network structure of the example database

Learning Phase (steps 1-3)

Perform causal Bayesian learning (illustration of Bayesian learning is given in Appendix) to the database instances in Table I to obtain the network structure as illustrated in Fig. 2.

There are two effect nodes in Bayes net of Fig. 2, that are, node D and node C. Both of them have two causal nodes. Therefore, the FDs detection phase follows the steps 4-7.

FDs Detection Phase (steps 4-7)

Check conditional probability tables for each causal edge, that is, AB → D and AD → C. Details of conditional probability values in each relation are shown in Tables II and III (combinations of attribute values that do not exist in the database table have been removed). Both dependency relationships hold. But they are composed of two causal nodes. We thus have to check other four dependencies: A→D, B→D, A→C, and D→C. The only dependency that holds is A→C, and its conditional probability table is shown in Table IV.

TABLE II. CONDITIONAL PROBABILITY FOR DEPENDENCY AB → D

	D=d1	D=d2	D=d3	D=d4
A=a1, B=b1	0.5	0.167	0.167	0.167
A=a1, B=b2	0.167	0.5	0.167	0.167
A=a2, B=b2	0.167	0.5	0.167	0.167
A=a2, B=b3	0.167	0.167	0.5	0.167
A=a3, B=b3	0.167	0.167	0.167	0.5

TABLE III. CONDITIONAL PROBABILITY FOR DEPENDENCY AD → C

	C=c1	C=c2
A=a1, D=d1	0.75	0.25
A=a1, D=d2	0.75	0.25
A=a2, D=d2	0.25	0.75
A=a2, D=d3	0.25	0.75
A=a3, D=d4	0.25	0.75

TABLE IV. CONDITIONAL PROBABILITY FOR DEPENDENCY A → C

	C=c1	C=c2
A=a1	0.833	0.167
A=a2	0.167	0.833
A=a3	0.25	0.75

Therefore, we can conclude that with the given example database as shown in Table I, the three discovered functional dependencies are: AB → D,
AD → C, and
A → C.

Example 2. The customer database instances are given as shown in Table V.

TABLE V. CUSTOMER DATABASE

A1	A2	A3	A4	A5	A6	A7
1	c1	Kitty	21000	Honda	31	Dang
1	c8	Kitty	21000	Toyota	41	Dum
1	c6	Kitty	21000	Nissan	51	Ple
2	c2	Somsak	20111	Mitsubishi	41	Dum
2	c5	Somsak	20111	Toyota	31	Dang
3	c4	Siri	19999	Toyota	31	Dang

Learning Phase (steps 1-3)

Perform causal Bayesian learning to the database instances in Table V to obtain the network structure as illustrated in Fig. 3.

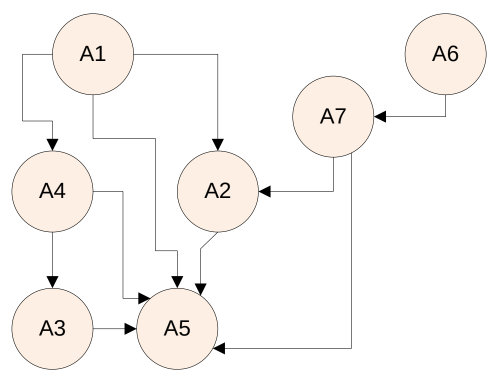

Figure 3. Bayesian network structure of customer database

There are five effect nodes in Bayes net of Fig. 3, that are, nodes A2, A3, A4, A5, and A7. Nodes A3, A4, and A7 have a single causal node, and A2 has two causal nodes. Thus, the FDs detection phase follows the steps 4-7, whereas the node A5 has to follow steps 8-10 because it has multiple causal nodes.

FDs Detection Phase (steps 4-7)

Single causal nodes (A3, A4, and A7) are easy to examine the dependencies as shown in Tables VI-VIII.

TABLE VI. CONDITIONAL PROBABILITY FOR DEPENDENCY A4 → A3

	A3=Kitty	A3=Somsak	A3=Siri
A4=21000	0.778	0.111	0.111
A4=20111	0.143	0.714	0.143
A4=1999	0.2	0.2	0.6

TABLE VII. CONDITIONAL PROBABILITY FOR DEPENDENCY A1 → A4

	A4=21000	A4=21000	A4=21000
A1=1	0.778	0.111	0.111
A1=2	0.143	0.714	0.143
A1=3	0.2	0.2	0.6

TABLE VIII. CONDITIONAL PROBABILITY FOR DEPENDENCY A6 → A7

	A7=Dang	A7=Dum	A7=Ple
A6=31	0.778	0.111	0.111
A6=41	0.143	0.714	0.143
A6=51	0.2	0.2	0.6

TABLE IX. CONDITIONAL PROBABILITY FOR DEPENDENCY A1A7 → A2

	A2=c1	A2=c2	A2=c4	A2=c5	A2=c6	A2=c8
A1=1, A7=Dang	0.375	0.125	0.125	0.125	0.125	0.125
A1=1, A7=Dum	0.125	0.125	0.125	0.125	0.125	0.375
A1=1, A7=Ple	0.125	0.125	0.125	0.125	0.375	0.125
A1=2, A7=Dang	0.125	0.125	0.125	0.375	0.125	0.125
A1=2, A7=Dum	0.125	0.375	0.125	0.125	0.125	0.125
A1=3, A7=Dang	0.125	0.125	0.375	0.125	0.125	0.125

It can be noticed from the conditional probability tables that the three dependencies A4→A3, A1→A4, and A6→A7 hold. The node A2 has two causal nodes: A1 and A7. The conditional probability of A1A7 → A2 is given in Table IX. It can be seen that all probability values are less than 0.5. Therefore, the dependency A1A7 → A2 does not hold, so do the dependencies A1 → A2 and A7 → A2.

The last examination is the dependency with multiple causal nodes A1A2A3A4A7 → A5. The steps 8-10 have to be applied. We then split the dependency relation into five cases: A1→ A5, A2→ A5, A3→ A5, A4→ A5, and A7→ A5. Conditional probabilities of the five cases are shown altogether in Table X. It is obviously seen that the only relation that holds is A2→ A5.

The discovered functional dependencies of database 2 are as follows:

A4→A3,
A1→A4,
A6→A7, and
A2→A5.

TABLE X. CONDITIONAL PROBABILITY FOR DEPENDENCY A1 → A5, A2 → A5, A3 → A5, A4 → A5, A7 → A5

	A5= Honda	A5= Toyota	A5= Nissan	A5= Mitsubishi
A1=1	0.3	0.3	0.3	0.1
A1=2	0.125	0.375	0.125	0.375
A1=3	0.167	0.5	0.167	0.167
A2=c1	0.5	0.167	0.167	0.167
A2=c2	0.167	0.167	0.167	0.5
A2=c4	0.167	0.5	0.167	0.167
A2=c5	0.167	0.5	0.167	0.167
A2=c6	0.167	0.167	0.5	0.167
A2=c8	0.167	0.5	0.167	0.167
A3=Kitty	0.3	0.3	0.3	0.1
A3=Somsak	0.125	0.375	0.125	0.375
A3=Siri	0.167	0.5	0.167	0.167
A4=21000	0.3	0.3	0.3	0.1
A4=20111	0.125	0.375	0.125	0.375
A4=1999	0.167	0.5	0.167	0.167
A7=Dang	0.3	0.3	0.3	0.1
A7=Dum	0.125	0.375	0.125	0.375
A7=Ple	0.167	0.5	0.167	0.167

V. EXPERIMENTATION

This section explains experimentation including steps in data preparation and parameter setting to learn Bayesian network structure with the WEKA software (http://www.cs.waikato.ac.nz/ml/weka). For data preparation, the customer database (Table V in example 2) has to be transformed into the *arff* format as follows:

```
@relation CustomerDatabase
@attribute A1  {1, 2, 3}
@attribute A2  {c1, c2, c4, c5, c6, c8}
@attribute A3  {kitty, somsak, siri}
@attribute A4  {21000, 20111, 1999}
@attribute A5  {honda, toyota, nissan, mitsubishi}
@attribute A6  {31, 41, 51}
@attribute A7  {dang, dum, ple }
@data
1, c1, kitty, 21000, honda, 31, dang
1, c8, kitty, 21000, toyota, 41, dum
1, c6, kitty, 21000, nissan, 51, ple
2, c2, somsak, 20111, mitsubishi, 41, dum
2, c5, somsak, 20111, toyota, 31, dang
3, c4, siri, 1999, toyota, 31, dang
```

After invoking WEKA and selecting the explorer task, click the classify tab and choose the ‘BayesNet’ algorithm for learning the Bayesian network structure. The default setting of this algorithm is inappropriate for learning the cause and effect structure, as required by this specific functional dependency application. We thus have to set the right parameter by clicking at the frame in which the name BayesNet appears (as pointed by the arrow in Fig. 4). A small window will be popped up. Then click the 'searchAlgorithm' option to choose the ICS search algorithm, and then click the OK button.

The Bayesian network structure can be visualized by right-clicking at the algorithm name below the 'Result list' panel (as shown in Fig. 5). Then choose 'Visualize graph.' A new pop-up window will appear as shown in Fig. 6.

Each node in the Bayesian network is associated with the conditional probability table. This table does not automatically display in the network structure, but it can be viewed by clicking at the node. The example of conditional probability table associated with node A7 is shown in Fig. 7.

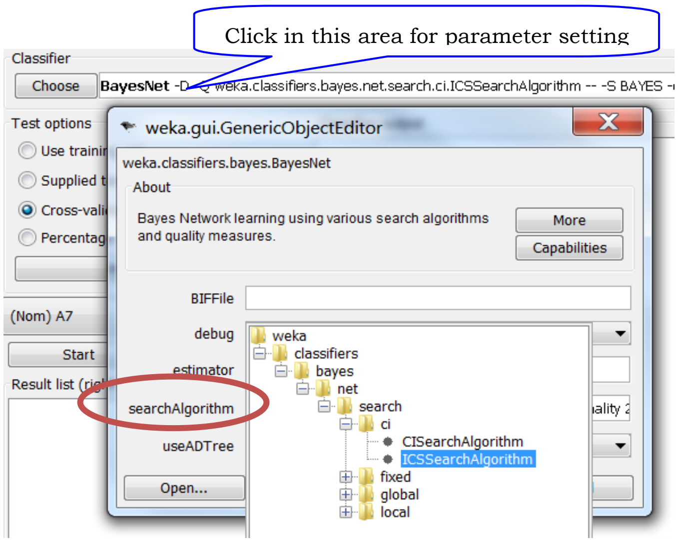

Figure 4. set the 'searchAlgorithm' parameter

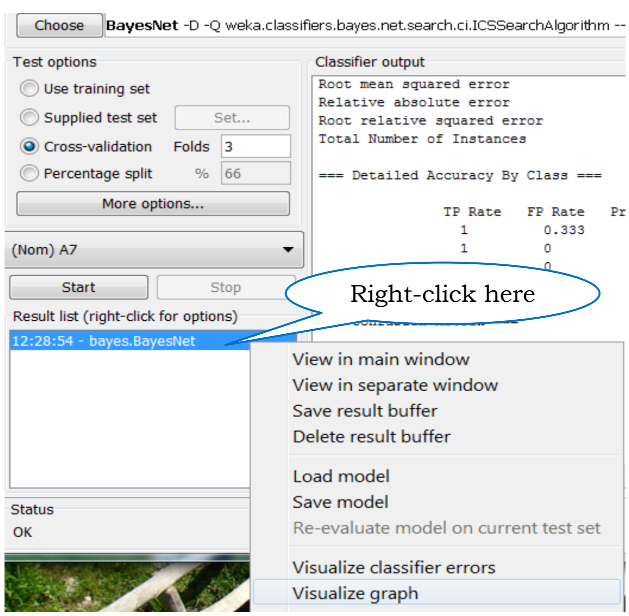

Figure 5. select 'Visualize graph' to view the Bayesian network

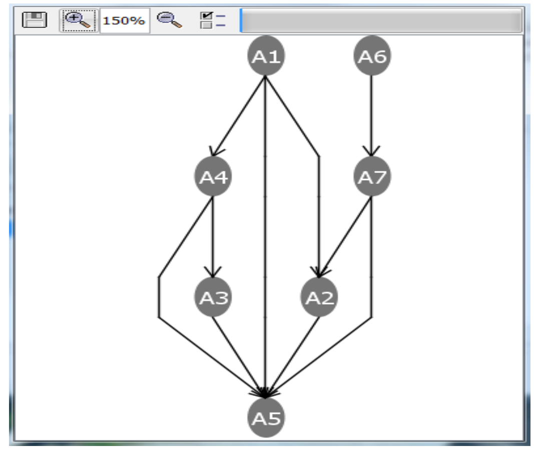

Figure 6. a Bayesian network structure for customer database

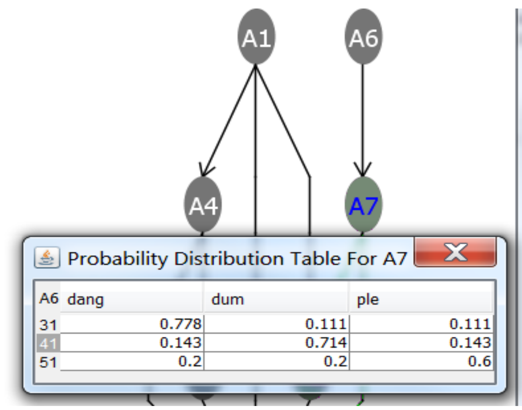

Figure 7. a conditional probability associated with each node

VI. CONCLUSION

The design of a complete database starts from the high-level conceptual design to capture detail requirements of the enterprise. Common tool normally used to represent these requirements is the entity-relationship, or ER, diagram and the product of this design phase is a conceptual schema. Typically, the schema at this level needs some adjustments based on the procedure known as normalization in order to reach a proper database design. Then, the database implementation moves to the lower abstraction level of logical design in which logical schema is constructed in a form of relations, or database tables.

In this paper, we propose a method to discover functional dependencies inherent in the conceptual schema from the database relation containing data instances. The discovering technique is based on the structure analysis of learned Bayes net. Heuristics are also applied on the relationship selection over the network structure.

The results from the proposed method are the same as the design schema obtained from the database designer. We plan to improve our methodology to discover a conceptual schema up to the level of multi-relations.

 ISBN: 978-1-61208-269-1

Acknowledgment

This research work has been funded by grants from the National Research Council of Thailand (NRCT) and Suranaree University of Technology through the full support of Data Engineering Research Unit, in which both authors are principal researchers.

References

[1] A. Alashqur, "Expressing database functional dependencies in terms of association rules," European Journal of Scientific Research, vol. 32, no. 2, 2009, pp. 260-267.

[2] P. Andritsos, R. J. Miller, and P. Tsaparas, "Information-theoretic tools for mining database structure from large data sets," in Proc. SIGMOD, 2004, pp. 731-742.

[3] W. W. Armstrong, "Dependency structures of database relationships," Information Processing, vol.74, 1974, pp. 580-583.

[4] J. Atoum, D. Bader, and A. Awajan, "Mining functional dependency from relational databases using equivalent classes and minimal cover," Journal of Computer Science, vol. 4, no. 6, 2008, pp. 421-426.

[5] R. R. Bouckaert, Bayesian Network Classifiers in Weka for Version 3-5-5, The University of Waikato, 2007.

[6] T. Calders, R. T. Ng, and J. Wijsen, "Searching for dependencies at multiple abstraction levels," ACM Transactions on Database Systems, vol. 27, no. 3, 2002, pp. 229-260.

[7] G. Chen, M. Ren, P. Yan, and X. Guo, "Enriching the ER model based on discovered association rules," Information Sciences, vol. 177, 2007, pp. 1558-1556.

[8] F. Chiang and R. J. Miller, "Discovering data quality rules," in Proc. VLDB, 2008, pp. 1166-1177.

[9] E. F. Codd, "A relational model of data for large shared data banks," Communications of the ACM, vol.13, no.6, 1970, pp. 377-387.

[10] C. J. Date and R. Fagin, "Simple conditions for guaranteeing higher normal forms in relational databases," ACM Transactions on Database Systems, vol.17, no.3, 1992, pp. 465-476.

[11] F. De Marchi, S. Lopes, and J. M. Petit, "Efficient algorithms for mining inclusion dependencies," in Proc. EDBT, 2002, pp. 464-476.

[12] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis, "Conditional functional dependencies for capturing data inconsistencies," ACM Transactions on Database Systems, vol. 33, no. 2, 2008, pp. 6:1-6:48.

[13] A. Heni, M.N. Omri, and A.M. Alimi, "Fuzzy knowledge representation based on possibilistic and necessary Bayesian networks," WSEAS Transactions on Information Science and Applications, vol.3, issue 2, 2006, pp. 224-231.

[14] T.Y. Hsieh and B.C. Kuo, "Information-based item selection with blocking strategy based on Bayesian network, WSEAS Transactions on Information Science and Applications, vol.7, issue 9, 2010, pp. 1160-1169.

[15] Y. Huhtala, J. Karkkainen, P. Porkka, and H. Toivonen, "Efficient discovery of functional and approximate dependencies using partitions," in Proc. ICDE, 1998, pp. 392-401.

[16] T. Hussain, "Schema transformations and dependency preservation," in Proc. 7th WSEAS Int. Conf. on Applied Informatics and Communications, Athens, Greece, August 24-26, 2007, pp. 313-318.

[17] R. S. King and J. J. Legendre, "Discovery of functional and approximate functional dependencies in relational databases," Journal of Applied Mathematics and Decision Sciences, vol. 7, no. 1, 2003, pp. 49-59..

[18] K. Lam and V. Lee, "Building decision trees using functional dependencies," in Proc. Int. Conf. Information Technology: Coding and Computing, 2004, p. 470.

[19] N. Lammari, I. Comyn-Wattiau, and J. Akoka, "Extracting generalization hierarchies from relational databases: A reverse engineering approach," Data & Knowledge Engineering, vol.63, 2007, pp. 568-589.

[20] H. Lee and C. Yoo, "A form driven object-oriented reverse engineering methodology," Information Systems, vol.25, no.3, 2000, pp. 235-259.

[21] J. Ma and K. Sivakumar, "Privacy-preserving Bayesian network parameter learning," WSEAS Transactions on Information Science and Applications, vol.3, issue 1, 2006, pp. 1-6.

[22] H. Mannila and K. J. Raiha, "Dependency inference," in Pro. VLDB, 1987, pp. 155-158.

[23] H. Mannila and K. J. Raiha, "On the complexity of inferring functional dependencies," Discrete Applied Mathematics, vol. 40, 1992, pp. 237-243.

[24] C. Mayfield, J. Neville, and S. Prabhakar, "ERACER: A database approach for statistical inference and data cleaning," in Proc. SIGMOD, 2010, pp. 75-86.

[25] R. E. Neapolitan, Learning Bayesian Networks, Pearson Education, 2004.

[26] N. Novelli and R. Cicchetti, "Functional and embedded dependency inference: a data mining point of view," Information Systems, vol. 26, 2001, pp. 477-506.

[27] N. Pannurat, N. Kerdprasop, and K. Kerdprasop, "Database reverse engineering based on association rule mining," International Journal of Computer Science Issues, vol. 7, no 2, March 2010, pp. 10–15.

[28] J. Perez, I. Ramos, V. Anaya, J. M. Cubel, F. Dominguez, A. Boronat, and J. A. Carsi, "Data reverse engineering for legacy databases to object oriented conceptual schemas," Electronic Notes in Theoretical Computer Science, vol.72, no.4, 2003, pp. 7-19.

[29] A. M. Silva and M. A. Melkanoff, "A method for helping discover the dependencies of a relation," in H. Gallaire, J. Minker, J. M. Nicolas (eds.), Advances in Data Base Theory, Plenum Press, 1981, pp. 115-133.

[30] M. Tuba and D. Bulatovic, "Design of an intrusion detection system based on Bayesian networks," WSEAS Transactions on Computers, vol.8, issue 5, 2009, pp. 799-809.

[31] T. Verma and J. Pearl, "An algorithm for deciding if a set of observed independencies has a causal explanation," in Proc. Conf. Uncertainty in Artificial Intelligence, 1992, pp. 323-330.

[32] C. Wyss, C. Giannella, and E. Robertson, "FastFDs: A heuristic-driven, depth-first algorithm for mining functional dependencies from relational instances," in Proc. DaWaK, 2001, pp. 101–110.

[33] H. Yao and H. J. Hamilton, "Mining functional dependencies from data," Data Mining and Knowledge Discovery, vol. 16, 2008, pp. 197-219.

 ISBN: 978-1-61208-269-1