Mining Functional Dependency from Relational Databases by Removing Redundant Candidates

Anupama A Chavan, Vijay Kumar Verma

Abstract— Discovery of functional dependencies from relational data base has been identified as an important database analysis technique. In this paper, we present a new approach for finding functional dependencies from large databases, based on partitioning the set of rows with respect to their attribute values. The discovery of functional dependencies is easy and efficient due to use of partitions, and the wrong or exceptional rows can be recognized easily. By using this we can eliminate equivalence attribute and redundant dependency. For standard databases the running times are better by several orders of degree over previously published results. The proposed algorithm is also works well for larger datasets than the previous methods.

Index Terms— Functional dependencies, closure of set, redundancy, normalization

I. INTRODUCTION

Normalization is the process of redesigning the database scheme to make it free from all anomalies [12]. Need of normalization is to get a good database design. The problems that occur with bad database design in the relational database are redundancy, updation anomaly, insertion anomaly, deletion anomaly. Repetition of information is called as redundancy which also leads to database inconsistency. Normalization breaks unstructured relation into smaller separate relations and then each individual relation is in normalized form. The idea behind decomposition is to eliminate redundant data and reduce data anomaly. There are many different levels of normalization depending on the purpose of database designer [3]

Redundancy is often caused by functional dependency. A functional dependency is a link between two sets of attributes in a relation. We can normalize a relation by removing unwanted FDs.

Functional dependencies are relationships between attributes of a relation; a functional dependency states that the value of an attribute is uniquely determined by the values of some other attributes. Automated database analysis is, of course, interesting for knowledge discovery and data mining (KDD) purposes and functional dependencies have applications in the areas of database management, reverse engineering and query optimization. [10]

II. BASIC CONCEPTS

A. Functional Dependency

Given a relation 'R', attribute 'Y' of 'R' is functional dependant on attribute 'X' of 'R' if- each 'X' value of "r' is associated with precisely one value of 'Y' in 'R' ".

Manuscript received October 01, 2011. (Fill the Details)

Miss Anupama A Chavan, Department Of Computer Science and Engineering, Lord Krishna College of Technology, Indore, MP, India.

MR. Vijay Kumar Verma, Department Of Computer Science and Engineering, Lord Krishna College of Technology, Indore, MP, India.

A functional dependency is a statement $X \rightarrow Y$ requiring that X functionally determines Y. For example city → state i.e. the state value depends on city value. [7, 8]

Properties of Functional Dependency

a. Let X and Y be candidates over a dataset D, if $X \rightarrow Y$ and $Y \rightarrow X$ hold, then X and Y are said to be equivalent candidates, denoted as $X \leftrightarrow Y$.
b. Let X, Y and Z are candidates over D. If $X \leftrightarrow Y$ and $XW \rightarrow Z$ hold, then $YW \rightarrow Z$ holds.
c. Let X, Y and Z be candidates over D. If $X \leftrightarrow Y$ and $WZ \rightarrow X$ hold, then $WZ \rightarrow Y$ holds.
d. Let F be a set of FDs over a dataset D and X be a candidate over D. The Closure of candidate X with respect to F, denoted Closure(X), is defined as {Y | X →Y can be deduced from F by Armstrong's axioms}. The nontrivial closure of candidate X with respect to F, denoted Closure'(X), is defined as Closure'(X) = Closure(X) – X.
e. Let $t_1, t_2, \ldots, t_n$ be all tuples in a dataset D, and X be a candidate over D. The partition over X, denoted $|\pi_x|$ is a set of the groups, such that t_i and t_j are in the same group if $t_i[X] = t_j[X]$. The number of the groups in the partition is called the cardinality of the partition, denoted $|\pi_x|$.

B. Agree Set

Let ti and tj be tuples and *X* an attribute set. The tuples ti and tj agree on X if ti[X] = tj [X]. The agree set of ti and tj is defined as follows:

$ag(ti,tj) = \{A \in R / ti[A]=tj[A]\}$. If r is a relation, $ag(r) = \{ag(ti,tj) / ti,tj \in r, ti \neq tj\}$.

C. Maximal Set

A maximal set is an attribute set X which, for some attribute A, is the largest possible set not determining A. We denote by max(dep(r),A) the set of maximal sets for A [4].

III. PREVIOUS RELATED WORK

In 1999 Yka, Juha, Pasi and Hannu presented "TANE", an efficient algorithm for finding functional dependencies from large database. To test the validity of FDs fast this algorithm is based on partitioning the set of rows with respect to their attribute values. TANE first makes use of partition method and equivalence class.[10]. In 2000 St_ephane Lopes, Jean-Marc Petit, and Lot_ Lakhal proposed a new efficient algorithm called Dep-Miner for discovering minimal non-trivial functional dependencies from large databases.[8] N. Novelli and R. Cicchetti, in 2001 proposed FUN: An Efficient Algorithm for Mining Functional and Embedded Dependencies. FUN describes FD approach at a general level only without detailing the optimizations due to the stripped partition database. Concept of free set is used for deriving FDs in this algorithm. [9]

In 2002 Hong Yao and Hamelton and But proposed "FD_Mine" algorithm. This algorithm needs three procedure one to discover Fd_Set other to discover EQ_Set and Prune Candidates and generate next level. Again in 2008 Hong Yao and Hamelton proposed an efficient rule discovery algorithm, called "FD_Mine", for mining functional dependencies from data by exploiting Armstrong's Axioms for functional dependencies.[6,7]. In 2008 Jalal Atoum, Dojanah Bader and Arafat Awajan proposed a new algorithm FD_Discover to discover FDs which utilizes the concepts of equivalent properties and minimal (Canonical) cover of FDs. The aim of this algorithm is to optimize the time requirements. [5]

In 2010 Y.V.Sreevani, Prof. T. Venkat Narayana Rao proposed Identification and Evaluation of Functional Dependency Analysis using Rough sets for Knowledge Discovery. A decision table based on certain factors and circumstances related to the knowledge base or the domain is used to discover the dependency between any subset of attributes using rough sets [4]. In 2011 Nittaya Kerdprasop and Kittisak Kerdprasop proposed Functional Dependency Discovery via Bayes Net Analysis. Proposed technique is based on the structure analysis of Bayesian network or Bayes net [3].

IV. DEP-MINER ALGORITHM

St_ephane Lopes, Jean-Marc Petit, and Lot_ Lakhal in 2000 proposed a new efficient algorithm called Dep-Miner. Dep-Miner is used for discovering agree sets, maximal sets, left-hand sides of minimal non-trivial functional dependencies and real-world Armstrong relations. In Dep-Miner the underlying idea is based on the concept of agree set, which groups all attributes having the same value for a given pair of tuples. Consider the simple employee database as shown in the below table.

I. Simple Employee Database

T No	Emp_No	Dep_No	Year	Dep_Name	Mgr_no
	A	B	C	D	E
1	1	1	2005	Production	5
2	1	5	2004	Marketing	12
3	2	2	2002	Sales	2
4	3	2	2008	Sales	2
5	4	3	2008	Purchase	2
6	5	1	1995	Production	5
7	6	5	1998	Marketing	12

So if we consider employee database given in table 1 maximal equivalence class is

MC= {{1,2},{1,6},{2,7},{3,4,5}}.

For building agree sets, we only consider couples of tuples belonging to a common equivalence class of MC. In our example the agree set for the pair of tuples(1,2) is ag{A} Similarly we have ag(1,6), ag(2,7) ag(3,4) ={B_D_E} ag(3,5)={E} ag(4,5)={C,E},so agree sets of r are ag(r)={A,BDE, E,CE}.

In Dep_Miner the underlying idea is based on the concept of agree set, which groups all attributes having the same value for a given pair of tuples. From these sets, maximal sets we can derive. The maximal sets for some attribute A are the largest possible sets of attributes not determining A. Then from the complements of these maximal sets they derive the lefthand sides of FDs using a levelwise algorithm for each attribute A it searches for lefthand sides X by increasing the size of X. The only step that requires accessing the database (or rather the stripped partition database) is the computation of agrees sets. This avoid computing agree sets for all pairs of tuples by limiting themselves to the tuples within MC_ the set of maximal equivalence classes of the stripped partition database $MC = max_{\subseteq}\{c \in \pi \mid \pi \in r\}$, MC= {{1,2},{1,6},{2,7},{3,4,5}}. An attribute A is included in the agree set of tuples $\{t_1,t_2\}$ if t_1 and t_2 belong to the same equivalence class in the stripped partition π_A. In our example the agree set for the pair of tuples(1,2) is ag{A} Similarly we have ag(1,6), ag(2,7) ag(3,4) ={B_D_E} ag(3,5)={E} ag(4,5)={C,E},so agree sets of r are ag(r)={A,BDE, E,CE}.

The maximal sets from the agree sets as follows, for an attribute A the maximal set $max(A, r) = max_{\subseteq}\{X \in ag(r) \mid A \notin X\}$. In this example, we have max (A, r)={BDE,CE} and the complement of the maximal set of A is cmax{A ,r}={AC, ABD}. The complete working process of Dep_Miner is shown in the table 2. [8]

II. Working of Dep-Miner Algorithm

RHS	cmax (RHS,r)	Size1 Candidate	Size1 Traversal	Size2 Candidate	Size2 Traversal
A	{AC,ABD}	A,B,C,D	A	BC,BD, CD	BC, CD
B	{BCDE,ABD,ABCD}	A,B,C,D, E	B,D	AC,AE, CE	AC, AE
C	{BCDE,AC, ABCD}	A,B,C,D, E	C	AB,AD, AE,DB, BE,DE	AB, AD, AE
D	{BCDE,ABD,ABCD}	A,B,C,D, E	B,D	AC,AE, CE	AC, AE
E	{BCDE}	B,C,D,E	B,C, D,E	-	

So final FDs are

BC→A, CD→A, D→B, AC→B, AE→B, AB→C, AD→C, AE→C, B→D, AC→D, AE→D, B→E, C→E, D→ E

V. PROBLEM STATEMENT

The problem addressed in this paper is to find functional dependencies among attributes in a database relation by removing redundant attribute. Specifically, we want to improve on previous proposed methods for this problem. Early methods for discovering of FDs were based on repeatedly sorting and comparing tuples to determine whether or not these tuples meet the FD definition. The disadvantage of this approach is that it does not utilize the discovered FDs as knowledge to obtain new knowledge If A →B has been discovered, a check is still made to determine whether or not AC→ B holds, by sorting on attributes AC and comparing on attribute B. Instead, AC→ B can be directly inferred from the previously obtained A→B without sorting and comparing tuples again. This approach is inefficient because of this extra sorting and because it needs to examine every value of the candidate attributes to decide whether or not a FD holds.

Consider the previous example of employee data base shown in table no 1. Form the Dep_Miner and FUN it is clear that functional dependency are generated for employee relational database are

BC→A AB→C B→D
CD→A AD→C B→E
AC→B AE→C C→E
AE→B AC→D D→E
D→B AE→D

Total dependencies 14 are generated in these dependencies some of the dependencies are redundant dependencies. Like B derives D and D derive B (B→D, D→B). So they are equal equivalent (B↔D). From the generated dependencies AB→C AD→C, B→E, D→E are redundant dependencies. Our approach is to remove these redundant dependencies and generate correct minimal dependencies. In the Dep_Miner algorithm traversal for size one, row for attribute A and B have the similar candidates and also same set of traversal. So we can take only attribute B and D can be deleted and replace attribute D by B. The steps of proposed method in shown in below figure.

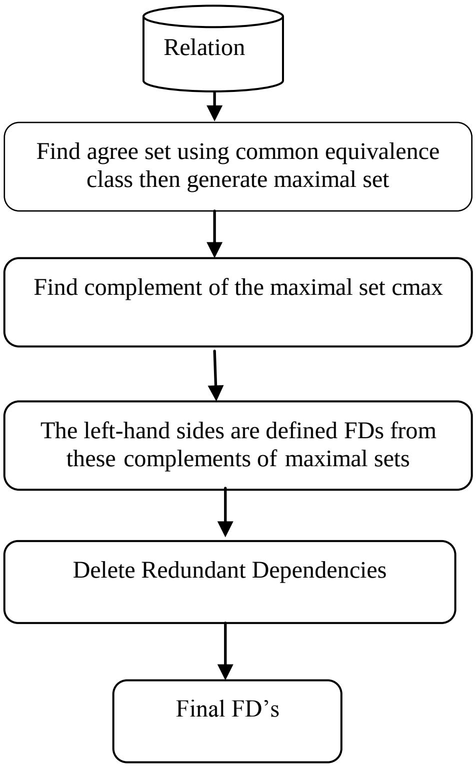

Fig 1-Steps in Proposed Algorithm

The complete working process of the proposed algorithm is shown in below table.

III. Working of Dep-Miner Algorithm

Now the final functional dependencies are

BC→A AB→C B→E
CB→A AE→C C→E
AC→B
AE→B

So there are only 8 functional dependencies are generated.

RHS	cmax(RHS,r)	Size1		Size2	
		Candidate	Traversal	Candidate	Traversal
A	{AC,ABD}	A,B,C,D	A	BC,BD,CD	BC,CD
B	{BCDE, ABD,ABCD}	A,B,C,D,E	B,D	AC,AE, CE	AC,AE
C	{BCDE, AC,ABCD}	A,B,C,D,E	C	AB,AD, AE,DB, BE,DE	AB,AD, AE
E	{BCDE}	B,C,D,E	B,C,D,E	-	-

VI. ALGORITHM

Discovering minimal functional dependencies

Input: a relation r

Output: minimal functional dependencies for relation r

1. AGREE SET: computes agree sets from r
2. CMAX SET: derives complements of maximal sets from agree sets
3. LEFT HAND SIDE: computes lhs of functional dependencies from complements of maximal sets
4. DELETE REDUNDANT DEPENDENCIES : find equivalence and remove them also replace deleted attribute by their equivalent
5. FD OUTPUT: outputs functional dependencies

VII. COMPARISON

We can compare number of dependencies at different level and the comparison chart is shown in the below table.

IV. Comparison of Method

Candidate	Dep_Miner	Proposed Algorithm
1	5	2
2	9	6
Total	14	8

The graph also shows the comparison with respect to number of dependencies generated by each of the method.

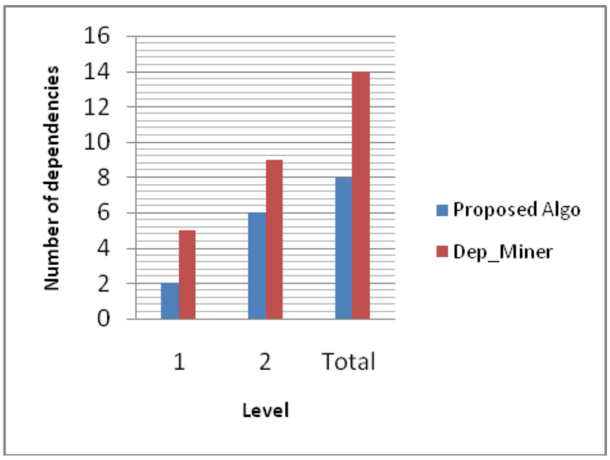

Fig 2-Comparision Graph

The submitting author is responsible for obtaining agreement of all coauthors and any consent required from sponsors before submitting a paper. It is the obligation of the authors to cite relevant prior work.

Authors of rejected papers may revise and resubmit them to the journal again.

VIII. CONCLUSION

The main benefit of proposed approach is that the DBA could use functional dependencies for normalizing existing relation schemas and also useful for better understanding relation schemas, and aiding to select only relevant functional dependencies among the whole (and possibly voluminous) set of extracted dependencies. The proposed approach not only reduce extra dependencies but also reduce execution time by deleting equivalence attribute which are unnecessary used in higher level generate. If we delete those attribute earlier then number of candidate at next level will automatically reduce this improves performance of the algorithm.

REFERENCES

[1] Jixue Liu, Jiuyong Li, Chengfei Liu, and Yong Feng Chen "Discover dependencies from Data—A review" IEEE transactions on knowledge and data engineering, vol. 24, no. 2, February 2012

[2] Vijaya Lakshmi, Dr. E. V. Prasad a fast and efficient method to find the conditional functional dependencies in databases International journal of engineering research and development e-issn: 2278-067, P-ISSN: 2278-800x, www.ijerd.com volume 3, issue 5 (august 2012), pp. 56

[3] Nittaya Kerdprasop and Kittisak Kerdprasop "Functional dependency discovery via Bayes net analysis" recent researches in computational techniques, non-linear systems and control ISBN: 978-1-61804-011

[4] Y.V.Sreevani, T. Venkat Narayana Rao "Identification and Evaluation of Functional Dependency Analysis using Rough sets for Knowledge Discovery " (IJACSA) International journal of advanced computer science and applications, vol. 1, no. 5, November 2010

[5] Jalal Atoum, Dojanah Bader and Larafat Awajan "Mining functional dependency from relational databases using equivalent classes and minimal cover " Journal of computer science 4 (6): 421-426, 2008 ISSN 1549-3636© 2008 science publications

[6] H. Yao, H.J. Hamilton and Cory J Butz "FD_Mine: Discovering Functional Dependencies in a database Using Eqivalences," J. Data Mining and Knowledge Discovery, vol. 16, no. 2, pp. 197-219, 2008

[7] H. Yao and H.J. Hamilton, "Mining Functional Dependenciesfrom Data," J. Data Mining and Knowledge Discovery, vol. 16, no. 2, pp. 197-219, 2008.

[8] St_ephane Lopes, Jean-Marc Petit, and Lot_ Lakhal "Dep-Miner Effective Discovery of Functional Dependencies and Armstrong Relations" Springer-Verlag Berlin Heidelberg 2000, pp. 350-364

[9] N. Novelli and R. Cicchetti, "Fun: An Efficient Algorithm for Mining Functional and Embedded Dependencies" Lecture Notes in Computer Science Volume 1973, 2001, pp 189-203

[10] Y. Huhtala, J. Karkkainen, P. Porkka, and H. Toivonen, "Tane : An Efficient Algorithm for Discovering Functional and Approximate Dependencies," Computer J., vol. 42, no. 2, pp. 100-111, 1999.

[11] Vijay Verma and Pradeep Sharma," Data Dependencies Mining In Database by Removing Equivalent Attributes" IJCSE, Vol.-1, Issue-1, July 2013

[12] Avi Silberschatz , Henry F. Korth ,S. Sudarshan,"Databse System Concepts, Sixth Edition, McGraw-Hill ISBN 0-07-352332-1