International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-6, August 2013

Functional Dependency Mining form Relational Database: A Survey

Anupama A Chavan, Vijay Kumar Verma

Abstract*— Data Mining represents the process of extracting interesting and previously unknown knowledge from data. Functional dependency plays a key role in database normalization. Normalization is process of rectifying database design to make sure that undesirable characteristics do not exist. To discover functional dependencies (FDs) from an existing relation instance is an important technique in data mining and database design. Functional dependencies are relationships between attribute of a database relation, a functional dependency state that the value of an attribute is uniquely determined by the values of some other attributes [5]. Functional dependency plays a key role in database normalization. Discovering FDs can also help a database designer to decompose a relational schema into several relations through the normalization process to get rid or eliminate some of the problems of unsatisfactory database design. In this paper we propose some of the existing methods and the techniques used by them.

Index Terms*— Decompose, Functional Dependencies, Instance, Normalization, Relations.

I. INTRODUCTION

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. Redundancy is the repetition of same information due to this memory space is wasted and it also leads to database inconsistency. As a result of redundancy updation anomaly occurs and also we cannot insert same information independently as thus insertion anomaly occurs. Deletion anomaly is inverse of insertion anomaly. Thus there should be some technique to make database free from all anomalies.

So Normalization is the process of redesigning the database scheme to make it free from all anomalies [12]. 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. There are following normal form that is used for database normalization are shown in figure 1.The main condition is 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 [3].

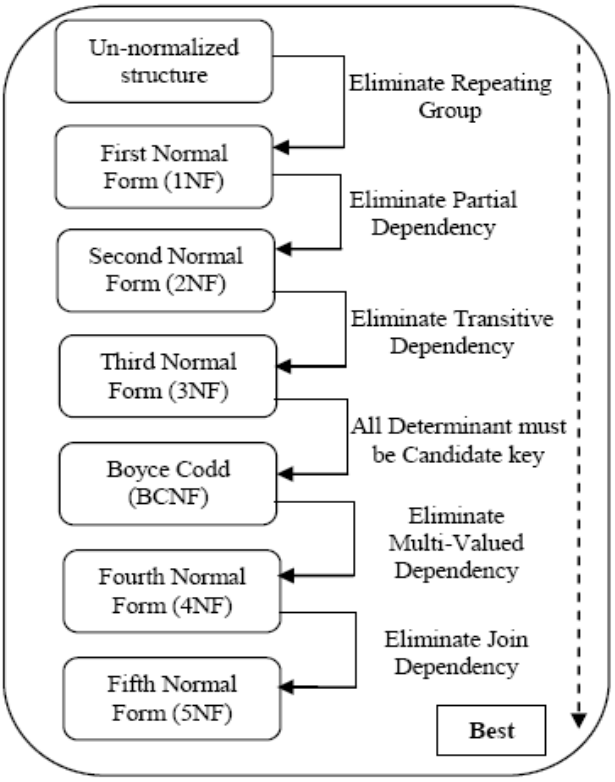

Fig 1-Steps of Normalization

II. FUNCTIONAL DEPENDENCY

To discover the dependencies that a database holds is the aim of dependency discovery. In research interests from the communities of database design dependency discovery has attracted a lot in machine learning and knowledge discovery [1]. Three types of dependencies are often involved in the discovery, functional dependencies (FDs), conditional functional dependence (CFDs) and inclusion dependencies (INDs) as shown in below figure 2.

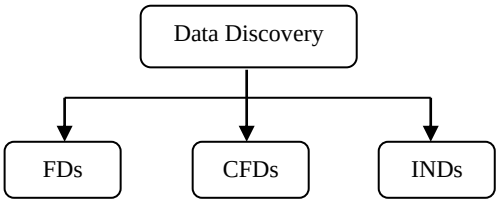

Fig 2-Types of data dependency

Manuscript published on 30 August 2013.
* Correspondence Author (s)
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.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

Functional dependency plays a key role in differentiating a good database design from a bad database design. A functional dependency is a type of constraint that is generalization of the notion of key.

Let R $\{A_1,\ldots\ldots\ldots\ldots,A_m\}$ be a database table schema and r be a set of tuples from $dom(A_1)\times\ldots\ldots\ldots\ldots\times dom(A_m)$ where dom(A) represents the domain of attribute A. The projection of a tuple t of r to a subset $X \subseteq R$ is denoted by t[X]. Similarly r[X] represents the projection of r to X. The number of tuples in the projection, called the cardinality, is denoted by |r[X]|. For simplicity, we often omit braces when the context is clear.

Definition of 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' " [12].

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

A. *Equivalent Attribute*

Let X and Y are 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. *Armstrong Augmentation and Transitivity Rule*

(1) Let X, Y and Z are candidates over D. If $X \leftrightarrow Y$ and $XW \rightarrow Z$ holds, then $YW \rightarrow Z$ holds.

(2) Let X, Y and Z are candidates over D. If X Y and $WZ \rightarrow X$ hold, then $WZ \rightarrow Y$ holds

C. *Nontrivial Closure*

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 $\{Y \mid X \rightarrow Y$ 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.

D. *Cardinality of the partition*

Let t1, t2, …, tn 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 $|\pi_x|$ [11].

III. FD DISCOVERY TECHNIQUES

Various algorithms for Functional dependency developed under each method are shown in below figure 3.

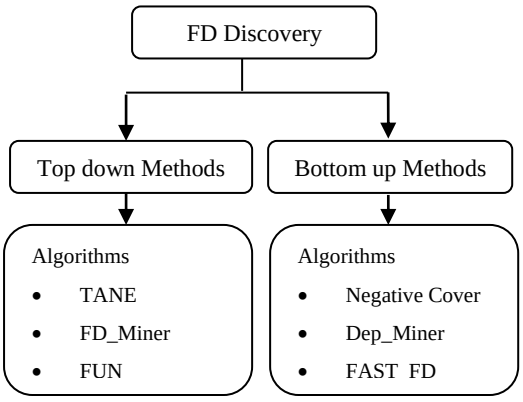

Fig 3 - Various Methods of FD Mining

Methods follow either top-down or bottom up approach. Top down method first generates candidate FDs and form an attribute lattice and the test there satisfaction. Then at lower level the satisfied FDs are used to prune candidate FDs to reduce the search space. Bottom up method compares the tuples of the relation to find agree-sets or different-sets. These sets are then used to derive FDs satisfied by the relation [1].

IV. RELATED WORK

Various methods have been presented for discovering FDs from large relational databases. Some of the techniques are as follows.

A. *TANE*

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. Then makes use of search technique to find all minimal non-trivial dependencies and prune the search space and then computing with the partitions. But TANE does repeatedly sorting and comparing of tuples to determine FDs which increases time complexity [10].

B. *Dep-Miner*

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. It combines the discovery of functional dependencies along with the construction of real-world Armstrong relations. In this method from the initial relation, a stripped partition database is extracted, using such partitions, agree sets are computed and thus, maximal sets are generated. On the one hand, they are used to build Armstrong relations. Their complements are derived then LHS of functional dependencies are computed easily [8].

C. *FUN*

In 2001 N. Novelli and R. Cicchetti, proposed Fun: An Efficient Algorithm for Mining Functional and Embedded Dependencies. In this algorithm Fds are derived using concept of free sets. A free set X is a set of attributes such that removing an attribute from X decrease the number of X's distinct values. Free sets correspond to LHS of FDs. To illustrate RHS of FDs, they define the closure and quasi-closure of an attribute set X. This approach relies heavily on counting the number of distinct values of attribute sets [9].

D. *FD_Mine*

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.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

This algorithm identifies equivalences among attributes, which can be used to reduce both the size of the dataset and the number of functional dependencies to be checked. This algorithm proves that the pruning does not lead to the loss of useful information. [6, 7]

E. FD_Discover

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.. In FD_Discover algorithm there is no need to check all attributes to discover functional dependency as a result of applying the equivalent properties. In FD_Discover algorithm only one procedure (Procedure Compute Minimal Nontrivial FD) is needed to discover immediately FD_Set, EQ_Set and pruning Candidate set for next level [5].

F. FDs Using Rough Sets

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. Due to the not consistent nature of the data, certain data values in the data table may be disagreeing. They suggested a method to solve this problem of data inconsistency based on the approach inspired by rough set. Rough set theory provides a collection of methods for extracting previously unknown data dependencies or rules from relational databases or decision tables. As established above it can be said that rough sets relates to entities databases, data mining, machine learning, and approximate reasoning etc [4].

G. FDs via Bayes Net Analyses

In 2011 Nittaya Kerdprasop and Kittisak Kerdprasop proposed Functional Dependency Discovery via Bayes Net Analysis. They proposed a novel technique to discover functional dependencies from the database table. The database designers can cover up inefficiencies inherent in their design with the help discovered dependencies. Proposed technique is based on the structure analysis of Bayesian network or Bayes net. Most data mining techniques applied to the problem of functional dependency discovery are rule learning and association mining. This work is a novel attempt of applying the Bayes net to this area of application. Search space is reduced if the FDs are discovered using this algorithm. Therefore, computational complexity is acceptable. The proposed method discovers 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. [3].

V. CONCLUSION

TANE, FD_Mine is based on partitioning the database and comparing the number of partitions. FD_Mine provides additional pruning rules. These pruning techniques are guaranteed not to eliminate any valid candidates, and whenever they are relevant, they reduce the size of the dataset or the number of checks required. FUN relies heavily on counting the number of distinct values of attribute sets. FD_Discover algorithm utilizes the concepts of equivalent properties and minimal (Canonical) cover of FDs. This algorithm is to optimize the time requirements. The FDs via Bayes Net Analysis is based on the structure analysis of learned Bayes net. Heuristics are also applied on the relationship selection over the network structure. FD using rough sets enables us to examine and to eliminate all unnecessary knowledge from the knowledge base by preserving only that part of the knowledge which is really useful. It gives some insight into rough sets which can be used to know data dependencies and extraction of knowledge. We represent a table showing various used in each method.

I. Techniques used FD Algorithm

FD discovery Algorithm	Technique used
TANE	Partitions
Dep-Miner	Difference Set
FUN	Embedded FD
FD_Mine	Partitions and Equivalences
FD_Discover	Equivalent Classes and Minimal Cover
FD Analysis using Rough sets	Rough sets
FD discovery by Bayes Net	Bayes Net and Heuristics

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

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.