The Survey: An Approaches for Mining Functional Dependencies in Relational Databases

Y.V. Dongre
Department of Computer Engineering
Vishwakarma Institute of Information Technology, Pune-48

Abstract- *This paper survey the mining of functional dependencies in relational database management system (RDBMS). In the data mining, discovery of functional dependencies (FDs) from an existing relation instance is an important technique which is helpful for designing of databases, specifically in the process of normalization. The most of methods covered reduces the FD discovery problem to that of finding FDs in relation, and then employs a different search strategies like level-wise or depth-first to determine minimal covers of newly discovered FDs . This work also studies previously developed algorithms for above problem for discovery of FDs, computations of minimal covers and compare these performances during their experimentations. In the database, time complexity of this type of search, insert and delete is O (n). This type of search is used in most of search applications where data is constantly inserting/ deleting to/from databases.*

Keywords- Data Mining, functional dependency, Relational Database, discovery.

I. INTRODUCTION

A dependency in database is relationship between attributes such that one cannot be computed until one or more other attributes have occurred in relation. A functional dependency is a type of constraint that is a generalization of the notion of key[1]. Functional dependencies (FDs) are the most important and widely accepted by database community as integrity constraints for relational databases. Study of functional dependency is an important part of relational database design and also plays important role in process of normalization. Functional dependency is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with attributes A and B, a functional dependency between the attributes is represented as A->B, which specifies B is functionally dependent on A. Here A is a determinant/determiner set and B is a dependent attribute. Each and every value of A is associated precisely with one B value.

Data mining provides methods that allow extracting from large data collections unknown relationships among the data items that are useful for decision making. Data mining is the process of discovering interesting patterns and knowledge from large amount of data [2]. The data sources can be on-line analytical processing (OLAP) and transaction data. Here we are applying data mining methodology in mining Functional dependencies.

The rest of paper is organized as follows: In Section 2, I write motivation for this work. I present literature survey on past work in Section 3, and the section 4 concludes the paper.

II. MOTIVATION

Motivations for surveying the mining FDs problem arise in several areas like data mining, database design, on-line transaction and analytical processing. Also this work is motivated by importance of functional dependency discovery. Since last decade, work on new approaches for discovery of FDs are not touched satisfactorily by database researchers.

III. SURVEY OF EXISTING RESEARCH

Many approaches are used to develop and implement efficient algorithms for mining FDs. Recently, Thorsten Papenbrock et.al.[3] in their experimental work describes, evaluates, and compares the seven most cited and most important algorithms, all solving problem of discovery of functional dependencies. They classified these algorithms into three different categories, explaining their commonalities. Then they described all algorithms with their main ideas. Their description is provided with additional details where the original papers were ambiguous or incomplete. Their evaluation of careful re-implementations of all algorithms spans a broad test space including synthetic and real-world data. They had shown that all functional dependency algorithms optimize for certain data characteristics and provides hints on when to choose which algorithm. Their experiments shown that FD discovery is still an open research field and none of the state-of-the-art algorithms in their experiments scales to datasets with hundreds of columns or millions of rows. Finally they claimed in their paper that all current approaches scale surprisingly poor hence showing potential for future research.

YKA HUHTALA et. al. [4] presented TANE, an efficient algorithm for finding functional dependencies from large databases. It is based on portioning the set of rows with respect to their attribute values, which makes testing the validity of functional dependencies fast even for large number of tuples. The use this approach also focuses on the discovery of approximate functional dependencies easy and efficient. An approximate functional dependency is a functional dependency that almost holds. For example speaking language is approximately determined by nationality. They also claim this method is erroneous and exceptional rows can be identified easily. This technique shown fast in practice during experimentations. But this work did not mention search criteria during traversal of tuples for discovery of FDs. Previous work by Yka Huhtala et. al. in [5], explained discovery of functional and approximate dependencies using partitions. They presented this approach for finding functional dependencies from large databases, based on partitioning mostly similar their above work.

Shyue-liang Wang et.al.[6] describes a data mining technique for determining approximate dependencies in similarity-relation-based fuzzy databases. Similarity-relation-based fuzzy data model is most suitable for describing analogical data over discrete domains, in addition to fuzzy-set-based fuzzy data models. Approximate dependency is an extension of functional dependency such that equality of tuples is extended and replaced with the notion of equivalence class. The approximate dependency we define here can capture more real-world integrity constraints then most existing functional dependencies on fuzzy databases. A level wise mining technique [5] is adopted in this work for the search of all possible nontrivial minimal approximate dependencies based on equivalence classes of attribute values.

Hong Yao [7], proposed a algorithm called FD-Mine. FD-Mine takes advantage of the rich theory of FDs to reduce both the size of the dataset and the number of FDs to be checked by using discovered equivalences. They had shown that the pruning does not lead to loss of information. They experimented on IS UCI datasets, and shown that FD-Mine can prune more candidates than older methods. Their results shown claims that the FD_Mine is valuable due to it reduces the size of the dataset but it does not lead to loss of information and eliminate any valid candidates.

An algorithm called FastFDs was introduced by Wyss C. et. al.[8], they employed a depth-first, heuristic driven search strategy for generating minimal covers of hyper graphs and computing minimal FDs from difference sets. This type of search is commonly used to solve search problems in Artificial Intelligence [9]. In that work, experimental results indicated that the level-wise strategy that is the hallmark of many successful data mining algorithms is in fact significantly surpassed by the depth-first, heuristic driven strategy. Here they indicated FastFDs is competitive for random integer valued instances of varying correlation factors, random Bernoulli instances, and real-life ML Repository relation instances. FastFDs employed because of the inherent space efficiency of the search. They revisited the comparison between Dep-Miner and TANE [3], including FastFDs. They reported several tests on distinct benchmark relation instances, comparing the Dep-Miner and FastFDs hyper graphs approaches to TANE's partitioning approach for mining FDs from a relation instance.

Heikki Mannila[10], explained more about need of theoretical framework in data mining and database design. A theoretical framework should be able to model typical data mining tasks (clustering, rule discovery, classification), be able to discuss the probabilistic nature of the discovered patterns and models, be able to talk about data and inductive generalizations of the data, and accept the presence of different forms of data. He written the framework should recognize that data mining is an interactive and iterative process, where comprehensibility of the discovered knowledge is important and where the user has to be in the loop, and that there is not a single criterion for what an interesting discovery is. During that work his favorite approach would be to combine the micro-economic view with inductive databases.

Stephane Lopes[11], propose a Dep-Miner which is an efficient algorithm for discovering minimal non-trivial functional dependencies from large databases. This approach combines the discovery of functional dependencies along with the construction of real-world Armstrong relations without additional execution time. These relations are small Armstrong relations taking their values in the initial relation. Discovering both minimal functional dependencies and real-world Armstrong relations facilitate the tasks of database administrators when maintaining and analyzing existing databases. They evaluated Dep-Miner performances by using a new benchmark database. Their experimental results shown both the efficiency of this approach compared to the best current algorithm TANE [4], and the usefulness of real-world Armstrong relations. This approach fits in a theoretical framework proposed in [12] for addressing the same issue. This work proposed solution by using techniques originated by data mining, which is provided with formal foundations ensuring the correctness the algorithms. The benefits of that approach is the dba is provided with two different representations. Functional dependencies used for normalizing existing relation schemas and real-world Armstrong relations are particularly useful for better understanding relation schemas.

A conditional functional dependencies (CFDs) is introduced by Philip Bohannon et. al. in [13], in that work they proposed a class of constraints and their applications in data cleaning. In contrast to traditional functional dependencies (FDs) that were developed mainly for schema design, CFDs aim at capturing the consistency of data by incorporating bindings of semantically related values. For example For CFDs they provided an inference system analogous to Armstrong's axioms for FDs, as well as consistency analysis. Since CFDs allow data bindings, a large number of individual constraints may hold on a table, complicating detection of constraint violations. They also developed techniques for detecting CFD violations in SQL as well as novel techniques for checking multiple constraints in a single query. They experimentally evaluated the performance of their CFD-based methods for inconsistency detection.

One more investigation about CFD was found in the work of Wenfei Fan et.al. in [14, 15] they investigated the discovery of conditional functional dependencies as an extension of functional dependencies (FDs) by supporting patterns of semantically related constants, and can be used as rules for cleaning relational data**.** In order identify data cleaning rules, they developed techniques for discovering CFDs from relations. They provided three methods for CFD discovery. The first is CFDMiner which is based on techniques for mining closed item sets and is used to discover constant CFDs. In the preprocessing task like data cleaning and data integration, constant CFDs are used through object identification. In their work they extended method for discovering FDs, CTANE algorithm which is levelwise, based on TANE[4] and another one is referred as FastCFD which is based on FastFD[8] is a depth-first approach algorithm. Their experimental study verified, CFDMiner can be multiple orders of magnitude faster than CTANE and FastCFD for constant CFD discovery. Also they claimed that CTANE works well when a given relation is large but it does not scale well with the arity of the relation.

Apart from most of the work surveyed on discovery of FDs, Jixue Liu et. al. in [16] described in details about discovery of inclusion and XML functional dependency. Inclusion dependencies support an essential semantics of the standard relational data model. An inclusion dependency is defined as the existence of attributes (side a) in a table A whose values must be a subset of the values of the corresponding attributes (side b) in another table B. When the side a term conforms a unique column or a primary key (PK) for the table B, the inclusion dependency is key-based. In this case, the side b term is a foreign key (FK) in A and the restriction is stated as A[FK] <<B[PK]. On the other hand, if the side b term does not constitute the key of A, the inclusion dependency is non-key-based. A XML functional dependency was briefly discussed by Loan T.H in [17], they proposed the XDiscover algorithm to detect a set of possible conditional functional fependencies (XCFDs) on a given XML data instance. Their approach employed the set of pruning rules to reduce the searching space and the number of XCFDs to be checked on the dataset. It can be used to enhance data quality management by suggesting possible rules and identifying non-compliant data. In [18] Bart Goethals et. al., shown their algorithm make use of the FDs to optimize the generation of frequent queries and prune redundant queries. They can also efficiently reduces the amount of queries generated.

IV. CONCLUSION

In this paper we surveyed and studied some well known algorithms concerned with discovery of FD in data mining and the methods for mining functional dependencies, conditional functional dependencies, approximate functional dependencies and inclusion functional dependencies in relational database schemes. The main problem in FDs mining is that it needs to scan relation sequentially and also computation of minimal covers. Many algorithms has been developed to solve the this problem as efficiently as possible. The depth-first search approach is more efficient than levelwise search in large databases. During this work, I observed performance comparison between algorithms of dependencies discovery. Under the FD mining techniques of data mining various algorithms namely: fastFD, fastCFD, Dep-Miner, FD-miner, CFDMiner, TANE and CTANE etc. algorithms has been studied.

REFERENCES

[1] Avi Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", Fifth Edition-2005

[2] Jiawei Han, Micheline Kamber and Jian Pei, "Data Mining: Concepts and Techniques", Third Edition-2011.

[3] Thorsten Papenbrock, Jens Ehrlich, Jannik Marten, Tommy Neubert, Jan-Peer Rudolph, Martin Schonberg, Jakob Zwiener and Felix Naumann, "Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms", Proceedings of VLDB 2015.

[4] Huhtala, Y., Karkkainen, J., Porkka, P., and Toivonen, H., (1999), TANE: An Efficient Algorithm for discovering Functional and Approximate Dependencies, The Computer Journal, V.42, No.20, pp.100-107.

[5] Huhtala, Y., Karkkainen, J., Porkka, P., and Toivonen,

H., Efficient Discovery of Functional and Approximate Dependencies Using Partitions, IEEE ICDE 1998.

[6] Shyue-liang Wang, Jenn-Shing Tsai and Been-Chian Chien, “Mining Approximate Dependencies Using Partitions on Similarity-relation-based Fuzzy Databases”, IEEE International Conference on Systems, Man and Cybernetics(SMC) 1999.

[7] Yao, H., Hamilton, H., and Butz, C., FD_Mine: Discovering Functional dependencies in a Database Using Equivalences, Canada, IEEE ICDM 2002.

[8] Wyss. C., Giannella, C., and Robertson, E. (2001), FastFDs: A Heuristic-Driven, Depth-First Algorithm for Mining Functional Dependencies from Relation Instances, Springer Berlin Heidelberg 2001.

[9] Russell, Stuart J. and Norvig, Peter. Artificial Intelligence: A ModernApproach. Prentice Hall, 1995.

[10] Mannila, H. (2000), Theoretical Frameworks for Data Mining, ACM SIGKDD Explorations, V.1, No.2, pp.30-32.

[11] Stephane Lopes, Jean-Marc Petit, and Lotfi Lakhal, “Efficient Discovery of Functional Dependencies and Armstrong Relations”, Springer 2000.

[12] Heikki Mannila and Kari-Jouko R¨aih¨a. Design by example: An application of Armstrong relations. Journal of Computer and System Sciences, 33(2):126{141, 1986.

[13] Philip Bohannon1 Wenfei Fan2,3 Floris Geerts3,4 Xibei Jia and Anastasios Kementsietsidis, “Conditional Functional Dependencies for Data Cleaning”, IEEE ICDE 2007.

[14] Wenfei Fan, Floris Geerts, Jianzhong Li, and Ming Xiong, “Discovering Conditional Functional Dependencies”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 5, 2011.

[15] Wenfei Fan, Floris Geerts, Laks V.S. Lakshmanan and, Ming Xiong, “Discovering Conditional Functional Dependencies”, IEEE International Conference on Data Engineering 2009.

[16] Jixue Liu, Jiuyong Li, Chengfei Liu, and Yongfeng Chen, “Discover Dependencies from Data—A Review”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 2,2012.

[17] Bart Goethals, Dominique Laurent and Wim Le, “Discovery and Application of Functional Dependencies in Conjunctive Query Mining”, Springer 2011.

[18] Loan T.H Vo, Jinli Cao and Wenny “Discovering Conditional Functional Dependencies in XML Data”, Australasian Database Conference (ADC) 2011.