International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.5, Sep-Oct. 2012 pp-3509-3511 ISSN: 2249-6645

Extraction of Conditional Functional Dependencies using Dataminig Techniques

Jetty Rajesh*,V.Kumar**
*M.Tech Student, Avanthi Institute of Engineering and Technology, Narsipatnam, India
**H.O.D, Dept of CSE, Avanthi Institute of Engineering and Technology, Narsipatnam, India

ABSTRACT: *Functional Dependencies (FD's) are recently extended as Conditional Functional Dependencies (CFD's) for cleaning Relational Data by supporting patterns of semantically related constants. Finding worth CFD's is an expensive process which needs more manual work. Using relations we are able to effectively identify data cleaning rules. Even the mining patter provided with new techniques, we in this paper provided three techniques for CFD discovery. First is a technique for closed mining item sets which is used for discovering constant CFD's ie's CFD's only with Patterns which Is essential to data cleaning and integration, which is referred to as CFDMiner. The other two Algorithms are CTANe and TANE which are developed for discovering General CFD's. And another Alogirhtmis FastCFD which is based on Depth-First Search Approach which will reduce the search space by identifying closed itemset mining. With our analysis, CFDMiner efficiently discovers constant CFDs. For general CFDs, CTANE works well when a given relation is large, but it does not scale well with the arity of the relation. FastCFD is far more efficient than CTANE when the arity of the relation is large; better still, leveraging optimization based on closed-itemset mining, FastCFD also scales well with the size of the relation. Thus we are providing the user to select set of cleaning rule discovery tools*

Keywords: *Privacy, Privelets, Data Publishing, Range count Queries.*

I. INTRODUCTION

This paper investigates the discovery of conditional functional dependencies (CFDs). CFDs are a recent extension of functional dependencies (FDs) by supporting patterns of semantically related constants, and can be used as rules for cleaning relational data. However, finding CFDs is an expensive process that involves intensive manual effort. To effectively identify data cleaning rules, we develop techniques for discovering CFDs from sample relations. We provide three methods for CFD discovery. The first, referred to as CFDMiner, is based on techniques for mining closed itemsets, and is used to discover constant CFDs, namely, CFDs with constant patterns only. The other two algorithms are developed for discovering general CFDs. The first algorithm, referred to as CTANE, is a levelwise algorithm that extends TANE, a well-known algorithm for mining FDs. The other, referred to as FastCFD, is based on the depthfirst approach used in FastFD, a method for discovering FDs. It leverages closed-itemset mining to reduce search space. Our experimental results demonstrate the following.

(a) CFDMiner can be multiple orders of magnitude faster than CTANE and FastCFD for constant CFD discovery. (b) CTANE works well when a given sample relation is large, but it does not scale well with the arity of the relation. (c) FastCFD is far more efficient than CTANE when the arity of the relation is large.

As remarked earlier, constant CFDs are particularly important for object identification, and thus deserve a separate treatment. One wants efficient methods to discover constant CFDs alone, without paying the price of discovering all CFDs. Indeed, as will be seen later, constant CFD discovery is often several orders of magnitude faster than general CFD discovery. Levelwise algorithms may not perform well on sample relations of large arity, given their inherent exponential complexity. More effective methods have to be in place to deal with datasets with a large arity. A host of techniques have been developed for (non-redundant) association rule mining, and it is only natural to capitalize on these for CFD discovery. As we shall see, these techniques can not only be readily used in constant CFD discovery, but also significantly speed up general CFD discovery. To our knowledge, no previous work has considered these issues for CFD discovery.

II. BACKGROUND & RELATED WORK

As remarked earlier, constant CFDs are particularly important for object identification, and thus deserve a separate treatment. One wants efficient methods to discover constant CFDs alone, without paying the price of discovering all CFDs. Indeed, as will be seen later, constant CFD discovery is often several orders of magnitude faster than general CFD discovery.

Levelwise algorithms may not perform well on sample relations of large arity, given their inherent exponential complexity. More effective methods have to be in place to deal with datasets with a large arity. A host of techniques have been developed for (non-redundant) association rule mining, and it is only natural to capitalize on these for CFD discovery. As we shall see, these techniques can not only be readily used in constant CFD discovery, but also significantly speed up general CFD discovery. To our knowledge, no previous work has considered these issues for CFD discovery.

In light of these considerations we provide three algorithms for CFD discovery: one for discovering constant CFDs, and the other two for general CFDs.

(Module: 1) we propose a notion of minimal CFDs based on both the minimality of attributes and the minimality of patterns. Intuitively, minimal CFDs contain neither redundant attributes nor redundant patterns. Furthermore, we consider frequent CFDs that hold on a sample dataset r, namely, CFDs in which the pattern tuples have a support in r above a certain threshold. Frequent CFDs allow us to accommodate unreliable data with errors and noise. Our algorithms find minimal and frequent CFDs to help users identify quality cleaning rules from a possibly large set of CFDs that hold on the samples.

(Module: 2) Our first algorithm, referred to as CFDMiner, is for constant CFD discovery. We explore the connection between minimal constant CFDs and closed and free patterns. Based on this, CFDMiner finds constant CFDs by leveraging a latest mining technique, which mines closed itemsets and free itemsets in parallel following a depth-first search scheme.

(Module: 3) Our second algorithm, referred to as CTANE, extends TANE to discover general CFDs. It is based on an attribute-set/pattern tuple lattice, and mines CFDs at level k + 1 of the lattice (i.e., when each set at the level consists of k+1 attributes) with pruning based on those at level k. CTANE discovers minimal CFDs only.

(Module: 4) Our third algorithm, referred to as FastCFD, discovers general CFDs by employing a depth-first search strategy instead of the levelwise approach. It is a nontrivial extension of FastFD mentioned above, by mining pattern tuples. A novel pruning technique is introduced by FastCFD, by leveraging constant CFDs found by CFDMiner. As opposed to CTANE, FastCFD does not take exponential time in the arity of sample data when a canonical cover of CFDs is not exponentially large.

(Module: 5) Our fifth and final contribution is an experimental study of the effectiveness and efficiency of our algorithms, based on real-life data (Wisconsin breast cancer and chess datasets from UCI) and synthetic datasets generated from data scraped from the Web. We evaluate the scalability of these methods by varying the sample size, the arity of relation schema, the active domains of attributes, and the support threshold for frequent CFDs. We find that constant CFD discovery (using CFDMiner) is often 3 orders of magnitude faster than general CFD discovery (using CTANE or FastCFD). We also find that FastCFD scales well with the arity: it is up to 3 orders of magnitude faster than CTANE when the arity is between 10 and 15, and it performs well when the arity is greater than 30; in contrast, CTANE cannot run to completion when the arity is above 17. On the other hand, CTANE is more sensitive to support threshold and outperforms FastCFD when the threshold is large and the arity is of a moderate size. We also find that our pruning techniques via itemset mining are effective: it improves the performance of FastCFD by 5-10 Folds and makes FastCFD scale well with the sample size. These results provide a guideline for when to use CFDMiner, CTANE or FastCFD in different applications.

These algorithms provide a set of promising tools to help reduce manual effort in the design of data-quality rules, for users to choose for different applications. They help make CFD-based cleaning a practical data quality tool.

III. SYSTEM DESIGN & ANALYSIS

This Component design diagram helps to model the physical aspects of an object oriented software system i.e., for the proposed framework it illustrates the architecture of the dependencies between service provider and consumer.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted

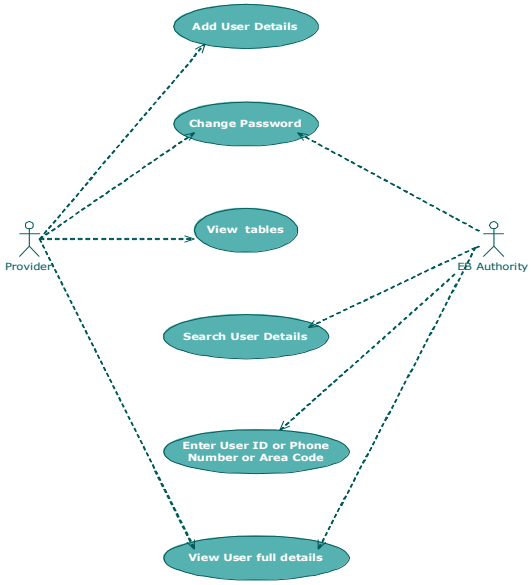

Fig.1: Inter-operational Use case diagram for the framework

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner

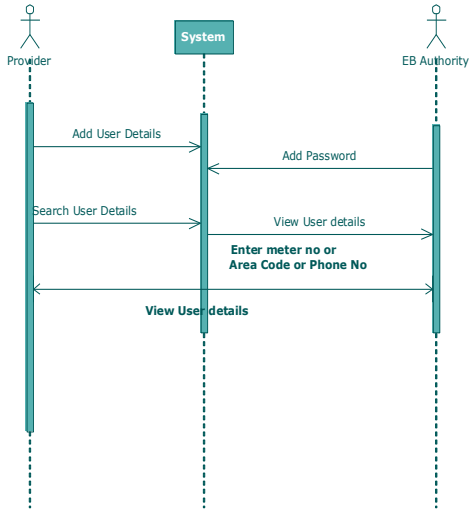

Fig.2: Inter-operational Sequence Diagram for the Framework

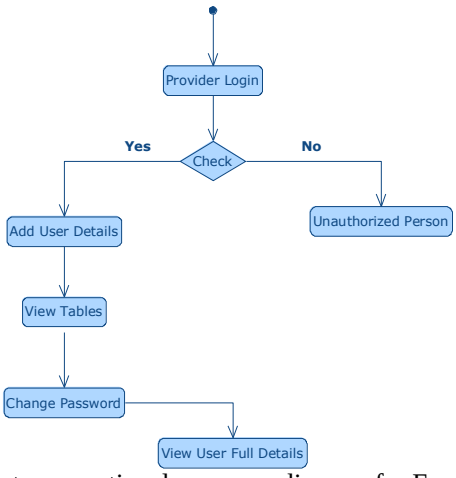

Fig.3: Inter-operational sequence diagram for Framework

RESULTS

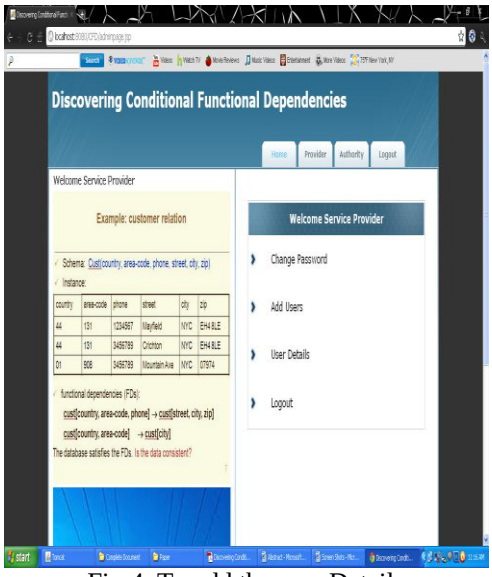

Fig.4: To add the user Details

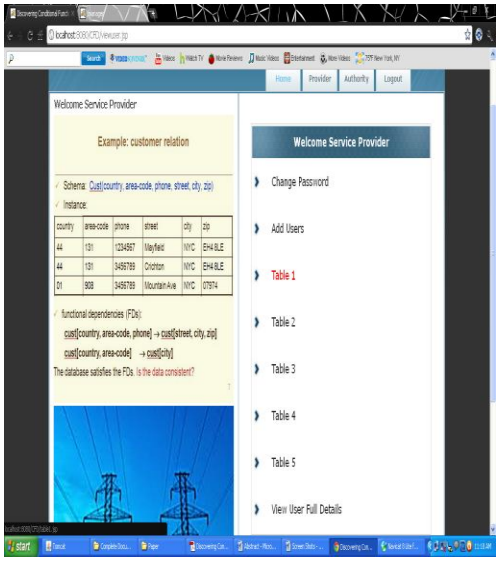

Fig.5: To give the Table information

Fig.6: User output Screen

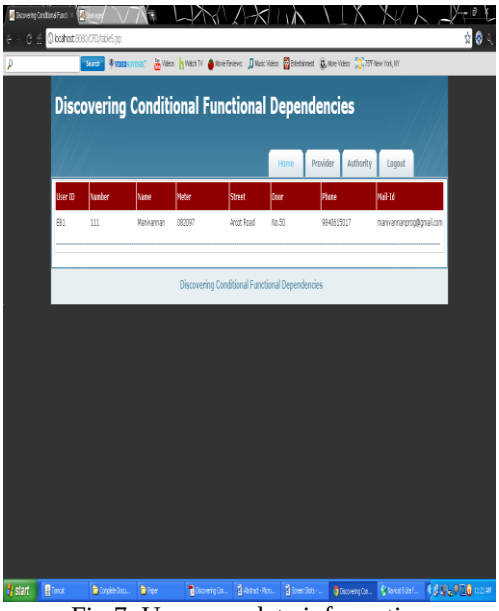

Fig.7: User complete information

CONCLUSION

We have developed and implemented three algorithms for discovering minimal CFDs: CFDMiner for mining minimal constant CFDs, a class of CFDs important for both data cleaning and data integration; CTANE for discovering general minimal CFDs based on the levelwise approach; and FastCFD for discovering general minimal CFDs based on a depth-first search strategy, and a novel optimization technique via closed-itemset mining. As suggested by our experimental results, these provide a set of tools for users to choose for different applications. When only constant CFDs are needed, one can simply use CFDMiner without paying the price of mining general CFDs. When the arity of a sample dataset is large, one should opt for FastCFD. When k-frequent CFDs are needed for a large k, one could use CTANE.

REFERENCE

[1] L. Bravo, W. Fan, F. Geerts, and S. Ma, “Increasing the expressivity of conditional functional dependencies without extra complexity,” in ICDE, 2008.

[2] G. Cormode, L. Golab, F. Korn, A. McGregor, D. Srivastava, and X. Zhang, “Estimating the confidence of conditional functional dependencies,” in SIGMOD, 2009.

[3] L. Bravo, W. Fan, and S. Ma, “Extending dependencies with conditions,” in VLDB, 2007.

[4] B. Goethals, W. L. Page, and H. Mannila, “Mining association rules of simple conjunctive queries,” in SDM, 2008.

[5] S. Lopes, J.-M. Petit, and L.Lakhal,“Efficient discovery of functional dependencies and armstrong relations,” in EDBT, 2000.

[6] T.Calders, R.T.Ng, and J.Wijsen, “Searching for dependencies at multiple abstraction levels,” TODS, vol. 27, no. 3, pp. 229–260, 2003.

[7] R. S. King and J. J. Legendre, “Discovery of functional and approximate functional dependencies in relational databases,” JAMDS, vol. 7, no. 1, pp. 49–59, 2003.

[8] I. F. Ilyas, V. Markl, P. J. Haas, P. Brown, and A. Aboulnaga, “Cords: Automatic discovery of correlations and soft functional dependencies,” in SIGMOD, 2004.

[9] H. Mannila and H.Toivonen, “Levelwise search and borders of theories in knowledge discovery,” Data Min. Knowl. Discov., vol. 1, no. 3, pp. 259–289, 1997.

[10] Gartner, “Forecast: Data quality tools, worldwide, 2006-2011,” 2007.

[11] S. Abiteboul, R. Hull, and V. Vianu, Foundations of Databases. Addison-Wesley, 1995.

[12] L. Golab, H. Karloff, F. Korn, D. Srivastava, and B. Yu, “On generating near-optimal tableaux for conditional functional dependencies,” in VLDB, 2008.