Discovering Functional Dependencies in Hidden Data using Approximate Formal Concept

Md. Abdullah Al Mamun
ID# mm1610594
Special Topic in Computing
Department of Computer Science & Engineering
Qatar University

Outline

- Introduction
- Objectives
- Background
- Related Works
- Methodology
- Experiment
- Result and Discussion
- Conclusion

Introduction

- Analyzing uncertain data is a challenging problem
- Data are rapidly increasing which complicates the analysis process
- Data reduction techniques handling uncertainty are highly required
- Display mostly related dataset to the users

Objective

- Conceptually reduce uncertain formatted data without losing dependencies between different attributes with respect to the original dataset

- Reduction method based on Formal Concept Analysis Theory (FCA) are proposed:
  - Approximate data reduction without loosing functional dependencies (FD)

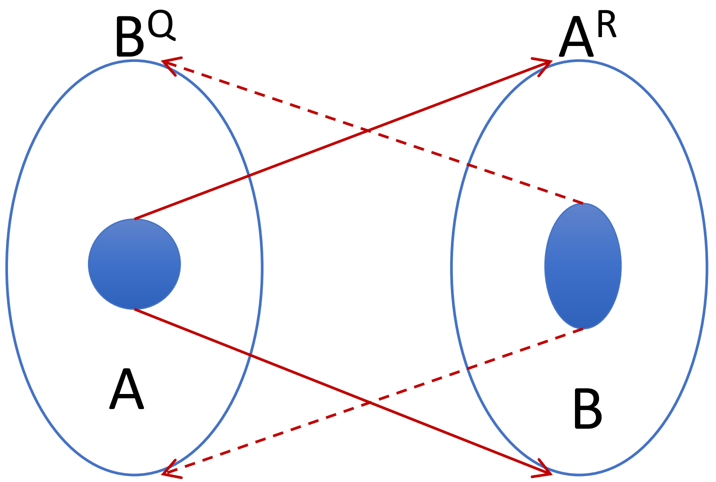
- To what extent the reduced dataset is preserving or even improving the functional dependency of a hidden database.

Background-Galois Connection

- Galois connection is a main notion in FCA used to extract implications
- Crisp Galois Connection operators R and Q are defined as:

$A^R = \{m \mid \forall g, g \in A: (g, m) \in I\}$, $B^Q = \{g \mid \forall m, m \in B: (g, m) \in I\}$

- Where $A \subseteq G$ and $B \subseteq M$



- Example

B

$A^R(B^Q)$

B^Q

	Preying	Flying	Bird	Mammal
Lion	1	0	0	1
Finch	0	1	1	0
Eagle	1	1	1	0
Hare	0	0	0	1
Ostrich	0	0	1	0

Flying → Bird

Selected Related Works

- Functional Dependency Discovery
  - *Papenbrock, Thorsten, et al. "Functional dependency discovery: An experimental evaluation of seven algorithms." Proceedings of the VLDB Endowment 8.10 (2015): 1082-1093.*
- Reduction
  - *Elloumi, Samir, et al. "A multi-level conceptual data reduction approach based on the Lukasiewicz implication." Information Sciences 163.4 (2004): 253-262.*
  - *Rezk, Eman, et al. "Uncertain training data set conceptual reduction: A machine learning perspective." Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on. IEEE, 2016.*

Methods – 7 Algorithms

The most cited and most important algorithm for functional dependency discovery

- The Tane algorithm by Huhtala et al.

- FUN by Novelli and Cic-chetti

  Traverses the attribute lattice level-wise bottom-up and applies partition refinement techniques to find functional dependencies.

- FD Mine by Yao et al

  Like Tane and Fun, Fd Mine traverses the attribute lattice level-wise bottom-up using stripped partitions and partition intersections to discover functional dependencies.

Cont.

- DFD by Abedjan et al

  It models the search space as a lattice of attribute combinations.

- Dep-Miner by Lopes et al

  Infers all minimal functional dependencies from sets of attributes that have same values in certain tuples.

- FastFDs by Wyss et al

  Improvement of Dep-Miner.

- FDEP by Flach and Savnik

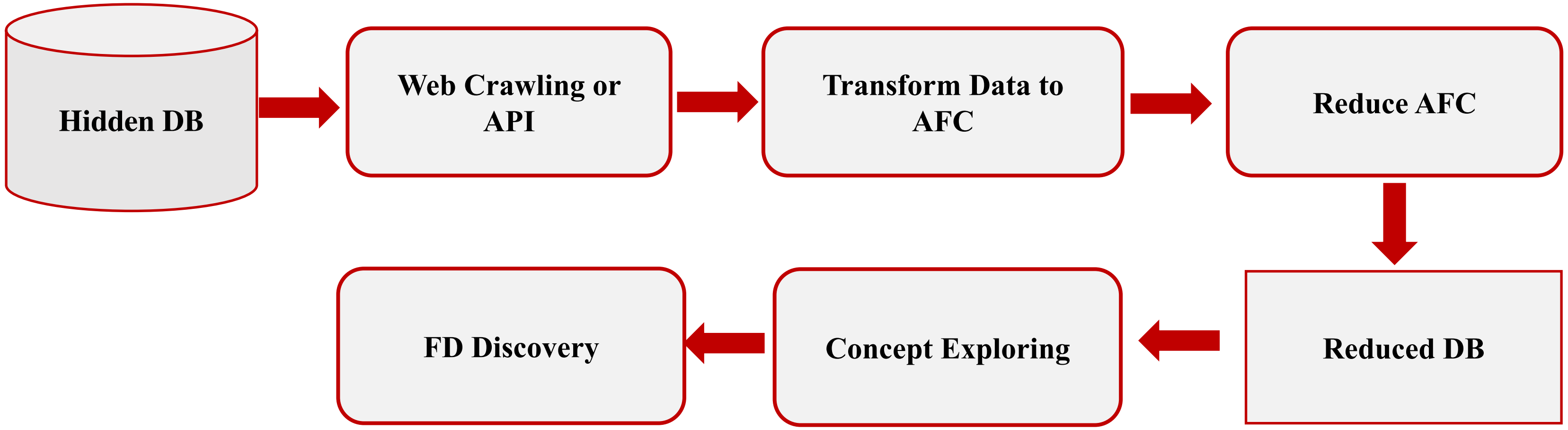
  Approach that is neither based on candidate generation nor on attribute set analysis.

New Approach

Approximate Formal Context Reduction (AFC): combining two research results gave rise of a new approach for data reduction without loss of functional dependencies

- Lukasiewicz data reduction algorithm applied for binary formal contexts [Elloumi et al 2004]
- Characterizing functional dependencies with formal concept analysis [Baixeries et al 2014]

Process Steps



Transform Data to Approximate FC

- Data transformed using similarity based pairwise comparison between tuples

$$Similarity = 1 - \frac{|n_1 - n_2|}{Max(n_1, n_2)}$$

- In the example, DBI is transformed to approximate FC with similarity threshold 0.7

➢ Similarity(T_1, T_2)(a)=1- $\frac{|1-4|}{Max(1,4)}$= 0.25

➢ Similarity(T_1, T_2)(b)=1- $\frac{|3-2|}{Max(3,2)}$= 0.67

➢ Similarity(T_1, T_2)(c)=1- $\frac{|4-5|}{Max(4,5)}$= 0.80

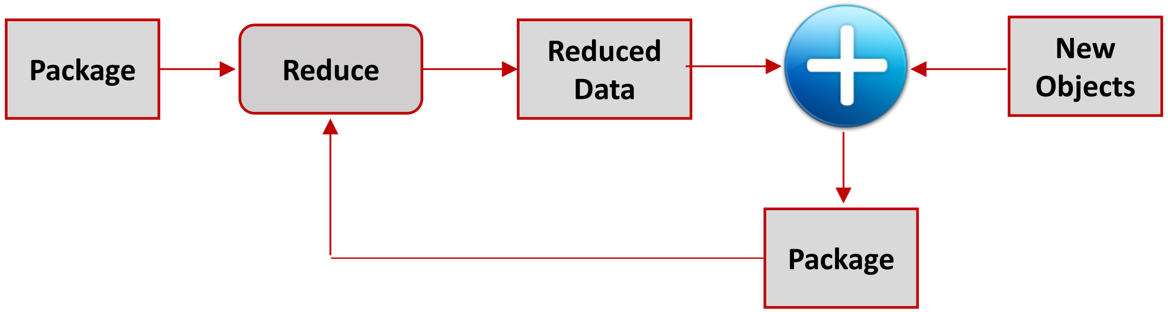
➢ Similarity(T_1, T_2)(d)=1- $\frac{|1-4|}{Max(1,4)}$= 0.25

Id	a	b	c	d
T_1	1	3	4	1
T_2	4	2	5	4
T_3	1	4	4	2
T_4	1	3	4	3

	a	b	c	d
(T_1, T_2)	0	0	1	0
(T_1, T_3)	1	1	1	0
(T_1, T_4)	1	1	1	0
(T_2, T_3)	0	0	1	0
(T_2, T_4)	0	0	1	1
(T_3, T_4)	1	1	1	0
(T_1, T_1)	1	1	1	1

Reduce FC

- The context is reduced using crisp incremental Lukasiewicz reduction (LR)
  - It works on packages of FC objects to avoid waiting the whole FC generation



  - It preserves FC implications that are equivalent to initial DB functional dependency

Crisp Incremental Lukasiewicz Data Reduction Algorithm

Input: Binary relation R, precision level δ =1
Output: Reduced relation RD
Begin
Initialize RD to R
For each object x in the domain of the remaining context RD, we do the following steps:

1. Find the set of properties P_X verifying object x
2. Find the set A of objects verifying the required values for the properties of x(P_X) at precision level $\boldsymbol{\delta}$
3. Let S_x=A-{x}, if the objects in S_x satisfy the same properties P_X at level $\boldsymbol{\delta}$ then remove x from RD

End for
End

Example

- Choosing $\delta = 0.7$ at X=O1

	A	B	C
O1	0.2	0.3	0.4
O2	0.5	0.7	1.0
O3	0.1	0.2	0.4
O4	0.4	0.3	1.0
O5	0.1	0.2	0.7
O6	0.2	1.0	0.4

	A	B	C
O1	0.2	0.3	0.4
O2	0.5	0.7	1.0
O3	0.1	0.2	0.4
O4	0.4	0.3	1.0
O5	0.1	0.2	0.7
O6	0.2	1.0	0.4

A={O2} $\boldsymbol{\delta}$ =0.7 Lukasiewicz Implication: min (1,1-a+b)

- A=min(1,1-0.2+0.5)= min(1,1.3)=1 > 0.7
- B=min(1,1-0.3+0.7)=min(1,1.4)=1 > 0.7
- C= min(1,1-0.4+1)=min(1,1.6)=1 > 0.7
- The three properties of O2 verified O1 at the $\boldsymbol{\delta}$ level
- Add O2 to the set of objects that verified O1 properties)
- A={O2}
- Now moving to O3

	A	B	C
O1	0.2	0.3	0.4
O2	0.5	0.7	1.0
O3	0.1	0.2	0.4
O4	0.4	0.3	1.0
O5	0.1	0.2	0.7
O6	0.2	1.0	0.4

A={O2} δ =0.7 Lukasiewicz Implication: min (1,1-a+b)

- A=min(1,1-0.2+0.1)= min(1,0.9)=0.9 > 0.7
- B=min(1,1-0.3+0.2)=min(1,0.9)=0.9 > 0.7
- C= min(1,1-0.4+0.4)=min(1,1)=1 > 0.7
- The three properties of O3 verified O1 at δ level=0.7
- Add O3 to the set of objects that verified O1 properties
- A={O2,O3}
- Now moving to O4

	A	B	C
O1	0.2	0.3	0.4
O2	0.5	0.7	1.0
O3	0.1	0.2	0.4
O4	0.4	0.3	1.0
O5	0.1	0.2	0.7
O6	0.2	1.0	0.4

A={O2,O3} δ=0.7 Lukasiewicz Implication: min (1,1-a+b)

- A=min(1,1-0.2+0.4)= min(1,1.2)=1
- B=min(1,1-0.3+0.3)=min(1,1)=1
- C= min(1,1-0.4+1.0)=min(1,1.6)=1
- The three properties of O4 implied O1 at δ level
- Add O4 to the set of objects that verified O1 properties
- A={O2,O3,O4}
- Now moving to O5

	A	B	C
O1	0.2	0.3	0.4
O2	0.5	0.7	1.0
O3	0.1	0.2	0.4
O4	0.4	0.3	1.0
O5	0.1	0.2	0.7
O6	0.2	1.0	0.4

A={O2,O3,O4} ***δ***=0.7 Lukasiewicz Implication: min (1,1-a+b)

- A=min(1,1-0.2+0.1)= min(1,0.9)=0.9
- B=min(1,1-0.3+0.2)=min(1,0.9)=0.9
- C= min(1,1-0.4+0.7)=min(1,1.3)=1
- The three properties of O5 implied O1 at ***δ*** level
- Add O5 to the set of objects that implied (verified O1 properties)
- A={O2,O3,O4,O5}
- Now moving to O6

	A	B	C
O1	0.2	0.3	0.4
O2	0.5	0.7	1.0
O3	0.1	0.2	0.4
O4	0.4	0.3	1.0
O5	0.1	0.2	0.7
O6	0.2	1.0	0.4

A={O2,O3,O4,O5} ***δ***=0.7 Lukasiewicz Implication: min (1,1-a+b)

- A=min(1,1-0.2+0.2)= min(1,1)=1
- B=min(1,1-0.3+1.0)=min(1,1.7)=1
- C= min(1,1-0.4+0.4)=min(1,1)=1
- The three properties of O6 verified O1 at ***δ*** level
- Add O6 to the set of objects that verified O1 properties
- A={O2,O3,O4,O5,O6}

Getting the Minimum along A

	A	B	C
O2	0.5	0.7	1.0
O3	0.1	0.2	0.4
O4	0.4	0.3	1.0
O5	0.1	0.2	0.7
O6	0.2	1.0	0.4

	A	B	C
O1	0.2	0.3	0.4
MINIMUM	0.1	0.2	0.4

- So *f*(A) = {A/0.1, B/0.2,C/0.4}

Getting the Minimum along A

	A	B	C
O1	0.2	0.3	0.4
MINIMUM	0.1	0.2	0.4

f(A) = {A/0.1, B/0.2,C/0.4} ***δ***=0.7 Lukasiewicz Implication: min (1,1-a+b)

- A=min(1,1-0.1+0.2)= min(1,1.1)=1 > 0.7
- B=min(1,1-0.2+0.3)=min(1,1.1)=1 > 0.7
- C= min(1,1-0.4+0.4)=min(1,1)=1 > 0.7
- The three properties of f(A) verified O1 at δ level
- So O1 can be removed normally with no change of Knowledge

Cont.

- Moving to the next Object O2
- Same $\boldsymbol{\delta}$=0.7

	A	B	C
O2	0.5	0.7	1.0
O3	0.1	0.2	0.4
O4	0.4	0.3	1.0
O5	0.1	0.2	0.7
O6	0.2	1.0	0.4

Experimentation

- Dataset:
- Abalone: The Abalone dataset contains the physical measurements of abalones, which are large, edible sea snails.
  - 4177 rows and 9 columns.
  - The columns include 1 categorical predictor (sex),
  - 7 continuous predictors
    - Length, Diameter, Height, Whole weight, Shucked weight, Viscera weight, Shell weight
- Tools: Java, Metanome, ConexExp
- Experimentation is also done for the DB: ncvoter and OLX

Results and Discussion

DB Name	Column	Row	25%	FD	50%	FD	75%	FD	100%	FD
Abalone	9	4177	56	50	64	50	65	54	66	59
ncvoter	19	1000	78	3752	113	4023	160	4035	174	4133
OLX	7	118	33	101	42	116	57	117	59	119

Conclusion

- Proposed AFC preserved the functional dependency
- Reduced FC done without losing any information
- Discovering Functional dependency is possible even if database is hidden

Acknowledgement

I would like to thank

- Dr. Ali Jaoua and his research team
- Especially Eng. Fahad Islam (RA)

for their kind support.

Thank You!