Week 3: Decision Reducts

CS286: Topics in Intelligent Systems

Katarzyna Tarnowska
Ph.D.

The Department of Computer Science
College of Science
San Jose State University

9/2/2020

Agenda

1. Introduction
   - Readings
   - Motivation
   - Case study
2. Definitions
   - The concept of a reduct
   - Discernibility relation
   - Decision reducts and the core
3. Algorithms
   - Discernability matrix-based
   - Activity
   - Heuristics-based

Agenda

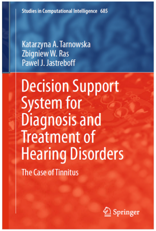
1. Introduction
   - Readings
   - Motivation
   - Case study
2. Definitions
   - The concept of a reduct
   - Discernibility relation
   - Decision reducts and the core
3. Algorithms
   - Discernability matrix-based
   - Activity
   - Heuristics-based

Agenda

1. Introduction
   - Readings
   - Motivation
   - Case study
2. Definitions
   - The concept of a reduct
   - Discernibility relation
   - Decision reducts and the core
3. Algorithms
   - Discernability matrix-based
   - Activity
   - Heuristics-based

Required readings

1. **Chapter 4.1.3: Reducts** in Decision Support System for Diagnosis and Treatment of Hearing Disorders. The Case of Tinnitus.
   OR
2. **Chapter 4.1: Decision Reducts** in Recommender System for Improving Customer Loyalty.



Agenda

1. Introduction
   - Readings
   - Motivation
   - Case study
2. Definitions
   - The concept of a reduct
   - Discernibility relation
   - Decision reducts and the core
3. Algorithms
   - Discernability matrix-based
   - Activity
   - Heuristics-based

Motivation

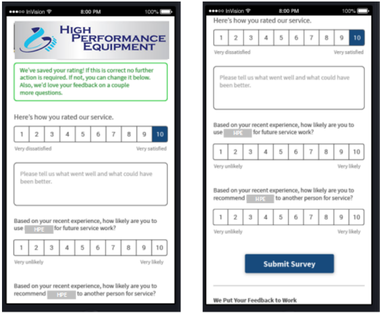
- Multidimensional datasets
- Performance issue for data mining
- Cognitive overload for analysts

Agenda

1. Introduction
   - Readings
   - Motivation
   - Case study
2. Definitions
   - The concept of a reduct
   - Discernibility relation
   - Decision reducts and the core
3. Algorithms
   - Discernability matrix-based
   - Activity
   - Heuristics-based

Case study - survey data

- Survey questions ("benchmarks") scored 1-10
- Each asks about customer experience in a particular area



Case study - large datasets of customer feedback surveys

- **Conditions**: survey answers
- **Decision**: net promoter score (NPS)

Client attributes			Customer attributes			Service attributes		Survey attributes and questions (customer experience on client's service)				NPS Status
ID	Name	Adress, ...	Name	Location	...	Time	Cost ,...	Q1 (score)	Q2 (score)	Q... (score)	QN (score)	Promoter
1												Passive
2												Detractor

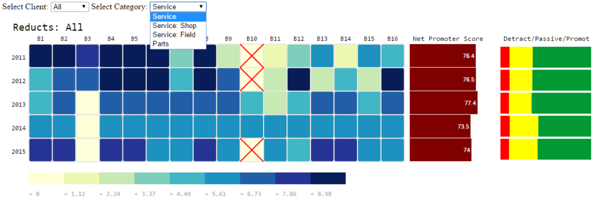
- Consulting developed a set of 200 questions possible to ask
- **Analytical problem**: which questions are *core* for insights into customer feedback?
- **Motivation**: optimize customer surveying time

Solution: compute *decision reducts*

- Turns out not all the attributes are necessary to retain the same knowledge
- In other words, you can discard certain attributes, without the loss of knowledge in the dataset

Solution: decision reducts + visual analytics

- Dataset as a “heatmap” (web-based visualization, using D3.js)



- Color-coding for the “criticality” of a question attribute
- Criticality computed as occurrence of an attribute in reducts

References: Tarnowska et al., 2017, Visual Analysis of Relevant Features in Customer Loyalty Improvement Recommendation in *Advances in Feature Selection for Data and Pattern Recognition or* 6.1: Decision Reducts as Heatmap in Tarnowska et al. *Recommender System for Improving Customer Loyalty (CLIRS)*

Agenda

1. Introduction
   - Readings
   - Motivation
   - Case study
2. Definitions
   - The concept of a reduct
   - Discernibility relation
   - Decision reducts and the core
3. Algorithms
   - Discernability matrix-based
   - Activity
   - Heuristics-based

Agenda

1. Introduction
   - Readings
   - Motivation
   - Case study
2. Definitions
   - The concept of a reduct
   - Discernibility relation
   - Decision reducts and the core
3. Algorithms
   - Discernability matrix-based
   - Activity
   - Heuristics-based

The concept of a reduct

- Consider a *decision table*
- Not every *attribute* is necessary for making a *decision*
- Only a subset of attributes is **essential** to make a decision
- A ***reduct*** - a **minimal** subset of attributes that retain the characteristics (knowledge) of the *entire* decision table

Agenda

Discernibility relation - definition

Definition

Let objects $x, y \in U$ and set of attributes $B \subset A$. We say that x, y are *discernible* by B when there exists $a \in B$ such that $a(x) \neq a(y)$.

x, y are *indiscernible by* B when they are identical on B, that is, $a(x) = a(y)$ for each $a \in B$.

$[x]_B$ denotes a set of objects *indiscernible with* x *by* B.

Discernibility relation - example

Indiscernibility: example 1

U	Headache	Muscle pain	Temp.	Flu
p1	Yes	Yes	Normal	No
p2	Yes	Yes	High	Yes
p3	Yes	Yes	Very-high	Yes
p4	No	Yes	Normal	No
p5	No	No	High	No
p6	No	Yes	Very-high	Yes
p7	No	Yes	High	Yes
p8	No	No	Very-high	No

B

Objects p_1, p_2, p_3 are **indiscernible** for attribute subset $B = \{Headache, Musclepain\}$
Also, there are three disjoint **indiscernibility classes**:

- $[p1]_B = \{p1, p2, p3\}$
- $[p4]_B = \{p4, p6, p7\}$
- $[p5]_B = \{p5, p8\}$

Agenda

Decision reduct - definition

Definition

A set of attributes $B \subset A$ is called **reduct of the decision table** if and only if:

- B keeps the discernibility of A, that is, for each $x, y \in U$, if x, y are discernible by A, then they are also discernible by B,
- B is irreducible, that is, none of its proper subset keeps discernibility properties of A (that is, B is *minimal* in terms of discernibility).

The set of attributes appearing in every reduct of information system A (decision table DT) is called **the core**.

An Example of Reducts & Core

U	*Headache*	*Muscle pain*	*Temp.*	*Flu*
U1	Yes	Yes	Normal	No
U2	Yes	Yes	High	Yes
U3	Yes	Yes	Very-high	Yes
U4	No	Yes	Normal	No
U5	No	No	High	No
U6	No	Yes	Very-high	Yes

Reduct1 = {Muscle-pain,Temp.}

U	*Muscle pain*	*Temp.*	*Flu*
U1,U4	Yes	Normal	No
U2	Yes	High	Yes
U3,U6	Yes	Very-high	Yes
U5	No	High	No

Reduct2 = {Headache, Temp.}

U	*Headache*	*Temp.*	*Flu*
U1	Yes	Norlmal	No
U2	Yes	High	Yes
U3	Yes	Very-high	Yes
U4	No	Normal	No
U5	No	High	No
U6	No	Very-high	Yes

CORE = {Headache,Temp} $\cap$ {MusclePain, Temp} = {Temp}

Agenda

1. Introduction
   - Readings
   - Motivation
   - Case study
2. Definitions
   - The concept of a reduct
   - Discernibility relation
   - Decision reducts and the core
3. Algorithms
   - Discernability matrix-based
   - Activity
   - Heuristics-based

Agenda

Discernability matrix-based algorithm - example 1

No	a	b	c	d
u1	*a0*	*b1*	*c1*	*y*
u2	*a1*	*b1*	*c0*	*n*
u3	*a0*	*b2*	*c1*	*n*
u4	*a1*	*b1*	*c1*	*y*

In order to discern equivalence classes of the decision attribute *d*, to preserve conditions described by the discernibility matrix for this table

	u1	u2	u3
u2	a,c		
u3	b	λ	
u4	λ	c	a,b

$C = \{a, b, c\}$

$D = \{d\}$

$$(a \vee c) \wedge b \wedge c \wedge (a \vee b)$$

$$= b \wedge c$$

Reduct = $\{b, c\}$

Discernability matrix-based algorithm - example 2

	a	b	c	d	E
u1	1	0	2	1	1
u2	1	0	2	0	1
u3	1	2	0	0	2
u4	1	2	2	1	0
u5	2	1	0	0	2
u6	2	1	1	0	2
u7	2	1	2	1	1

	u1	u2	u3	u4	u5	u6
u2	λ					
u3	b,c,d	b,c				
u4	b	b,d	c,d			
u5	a,b,c,d	a,b,c	λ	a,b,c,d		
u6	a,b,c,d	a,b,c	λ	a,b,c,d	λ	
u7	λ	λ	a,b,c,d	a,b	c,d	c,d

$Core = \{b\}$

$Reduct1 = \{b,c\}$

$Reduct2 = \{b,d\}$

Discernability function:

(b+c+d)b(a+b+c+d)(b+c)(b+d)(a+b+c)(a+b)(c+d) = b(c+d) = bc + bd

Discernability matrix-based algorithm - example 3

Information System

	a	b	c	d	f
x1	0	L	0	L	0
x2	0	R	1	L	1
x3	0	L	0	L	0
x4	0	R	1	L	1
x5	1	R	0	L	2
x6	1	R	0	L	2
x7	2	S	2	H	3
x8	2	S	2	H	3

REDUCTS

Discernibility Matrix

x1								
x2	bc							
x3	-	bc						
x4	bc	-	bc					
x5	ab	ac	ab	ac				
x6	ab	ac	ab	ac	-			
x7	abcd	abcd	abcd	abcd	abcd	abcd		
x8	abcd	abcd	abcd	abcd	abcd	abcd	-	
	x1	x2	x3	x4	x5	x6	x7	x8

Discernibility Function:

f(a, b, c, d) = (b + c) (a + b) (a + b + c + d) (a + c) =(b + c) (a + b) (a + c) =
(ba + bb + ca + cb) (a + c) = (b + ca) (a + c)
ba + bc + ca

Reducts: {b, a}, {c, a}, {c, b}

(b=L) → (f=0);
(a=0)*(b=R) → (f=1);
......

Discernability matrix-based algorithm to find reducts

- Construct a discernability matrix from a decision table
- Construct a discernability function from a discernability matrix
- The function is in conjunctive normal form (CNF)
- To convert to DNF (disjunctive normal form), apply the logic *law of absorption*: $a \wedge (a \vee b) = a$
- Find *prime implicants* from DNF
- These are the *reducts*

Prime implicants: example

Discernability function in conjunctive normal form (CNF):

$$\Phi^U = (p_1^a \vee p_1^b \vee p_2^b) \wedge (p_1^a \vee p_2^a \vee p_3^b)$$
$$\wedge (p_1^a \vee p_2^a \vee p_3^a)$$
$$\wedge (p_2^a \vee p_3^a \vee p_1^b) \wedge (p_2^a \vee p_2^b \vee p_3^b)$$
$$\wedge (p_2^a \vee p_3^a \vee p_4^a \vee p_1^b \vee p_2^b \vee p_3^b)$$
$$\wedge (p_3^a \vee p_4^a) \wedge (p_4^a \vee p_3^b) \wedge (p_2^a \vee p_1^b)$$
$$\wedge (p_2^b \vee p_3^b) \wedge (p_3^a \vee p_2^b) \wedge p_2^b.$$

Discernability function in disjunctive normal form (DNF):

$$\Phi^U = (p_2^a \wedge p_4^a \wedge p_2^b) \vee (p_2^a \wedge p_3^a \wedge p_2^b \wedge p_3^b)$$
$$\vee (p_3^a \wedge p_1^b \wedge p_2^b \wedge p_3^b) \vee (p_1^a \wedge p_4^a \wedge p_1^b \wedge p_2^b).$$

Here, we have found four *prime implicants*

$\{p_2^a, p_4^a, p_2^b\}$ is the optimal result, because it is the minimal subset of P.

Agenda

Programming assignment 1

Implement an algorithm for finding decision reducts

1. Read in data in .csv
2. Convert to decision table (DT)
3. Create discernability matrix from DT
4. Construct discernability function from discernability matrix
5. Convert discernability function from CNF to DNF
6. Find prime implicants
7. Print out reducts

Testing the algorithm:

- Test on a toy example (i.e. from the lectures)
- Test on a large dataset (i.e. from the lab)

Agenda

1. Introduction
   - Readings
   - Motivation
   - Case study
2. Definitions
   - The concept of a reduct
   - Discernibility relation
   - Decision reducts and the core
3. Algorithms
   - Discernability matrix-based
   - Activity
   - Heuristics-based

Other approaches - finding a core using heuristics

- The number of possible reducts can be 2^{N-1} where N is the number of attributes.
- Selecting the optimal reduct from all of possible reducts is time-complex and heuristics must be used.

Self-check

Make sure you know:

1. The definition of the *discernibility relation*.
2. The definition of *decision reduct*.
3. An example/case study of a dataset where finding decision reducts would be beneficial and justify the case.
4. How to apply the discernability matrix-based algorithm for finding reducts in a sample information system.