Rough sets in Discretization

Nguyen Hung Son

This presentation was prepared on the basis of the following public materials:

1. Jiawei Han and Micheline Kamber, „Data mining, concept and techniques" http://www.cs.sfu.ca
2. Gregory Piatetsky-Shapiro, „kdnuggest", http://www.kdnuggets.com/data_mining_course/

Outline

- Classification of discretization methods
- Rough set and Boolean approach to discretization
  - Problem encoding
  - MD-Heuristics
  - Properties of MD heuristics

Classification of discretization methods

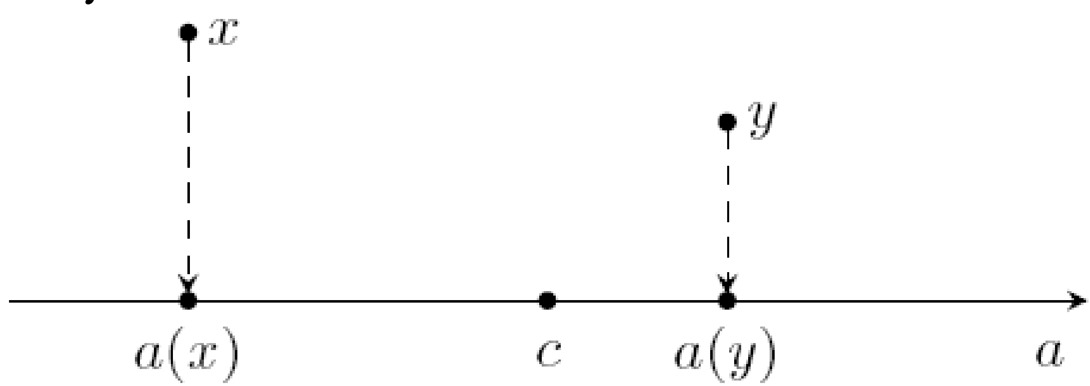
1. **Local versus Global methods:**
   - Local methods produce partitions that are applied to localized regions of object space (e.g. decision tree).
   - Global methods produce a mesh over k-dimensional real space, where each attribute value set is partitioned into intervals independent of the other attributes.
2. **Static versus Dynamic Methods:**
   - Static methods perform one discretization pass for each attribute and determine the maximal number of cuts for this attribute independently of the others.
   - Dynamic methods are realized by searching through the family of all possible cuts for all attributes simultaneously.
3. **Supervised versus Unsupervised methods**:
   - *Unsupervised methods* do not make use of decision values of objects
   - *Supervised methods* utilize the decision attribute in discretization process.

Discernibility by cuts

- Let S = (U,A [{d}) be a given decision table.
- We say that a cut $(a; c)$ on an attribute *a* **discerns** a pair of objects (x, y) if

$$(a(x) - c)(a(y) - c) < 0$$

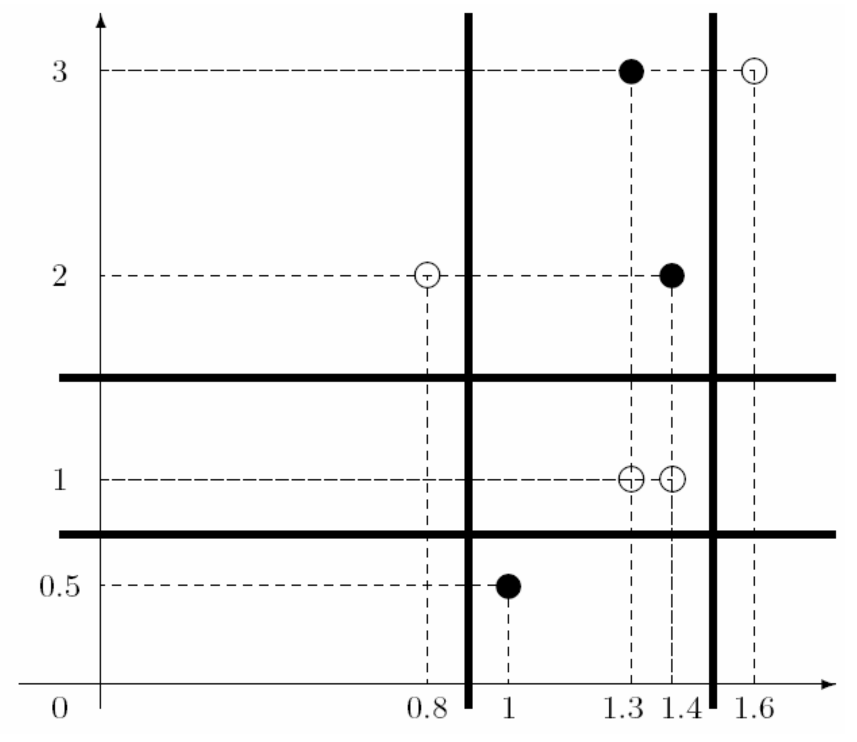
- Two objects are discernible by a set of cuts **C** if they are discernible by at least one cut from **C**.



Consistent set of cuts

- A set of cuts C is consistent with **S** (or **S** -consistent, for short) if and only if for any pair of objects (x, y) such that $dec(x) \neq dec(y)$, the following condition holds:

  IF x, y are discernible by A **THEN** x, y are discernible by **C**



Optimal discretization problem

OptiDisc : optimal discretization problem
input: A decision table $\mathbb{S}$.
output: $\mathbb{S}$-optimal set of cuts.

DiscSize : k-cuts discretization problem
input: A decision table $\mathbb{S}$ and an integer k.
question: Decide whether there exists a $\mathbb{S}$-irreducible set of cuts $\mathbf{P}$ such that $card(\mathbf{P}) < k$.

Theorem 2 (Computational complexity of discretization problems).

1. DiscSize *is* NP*-complete.*
2. OptiDisc *is* NP*-hard.*

Boolean reasoning approach to discretization

- Boolean variable
- Encoding function
- MD heuristics

Boolean variable

- **C** – a set of candidate cuts defined either
  - by an expert/user or
  - by taking all generic cuts
- We associate with each cut $(a,c) \in \mathbf{C}$ a Boolean variable $p_{(a,c)}$
- $p_{(a,c)} = 1 \quad \Leftrightarrow$ the cut (a,c) is selected

Encoding function

- For any pair of objects $u_i, u_j \in U$.

$$\mathbf{X}_{i,j}^{a} = \{(a, c_k^a) \in \mathbf{C}_a : (a(u_i) - c_k^a)(a(u_j) - c_k^a) < 0\}.$$

$$\mathbf{X}_{i,j} = \bigcup_{a \in A} \mathbf{X}_{i,j}^{a}$$

- Discernibility function for two objects

$$\psi_{i,j} = \begin{cases} \Sigma \mathbf{x}_{i,j} & \text{if } \mathbf{X}_{i,j} \neq \emptyset \\ 1 & \text{if } X_{i,j} = \emptyset \end{cases}$$

- Discernibilitv function for discretization problem

$$\Phi_{\mathbb{S}} = \prod_{d(u_i) \neq d(u_j)} \psi_{i,j}.$$

Example: Boolean variables

$\mathbb{S}$	a	b	d
u_1	0.8	2	1
u_2	1	0.5	0
u_3	1.3	3	0
u_4	1.4	1	1
u_5	1.4	2	0
u_6	1.6	3	1
u_7	1.3	1	1

$$a(U) = \{0.8, 1, 1.3, 1.4, 1.6\};$$
$$b(U) = \{0.5, 1, 2, 3\},$$

a:

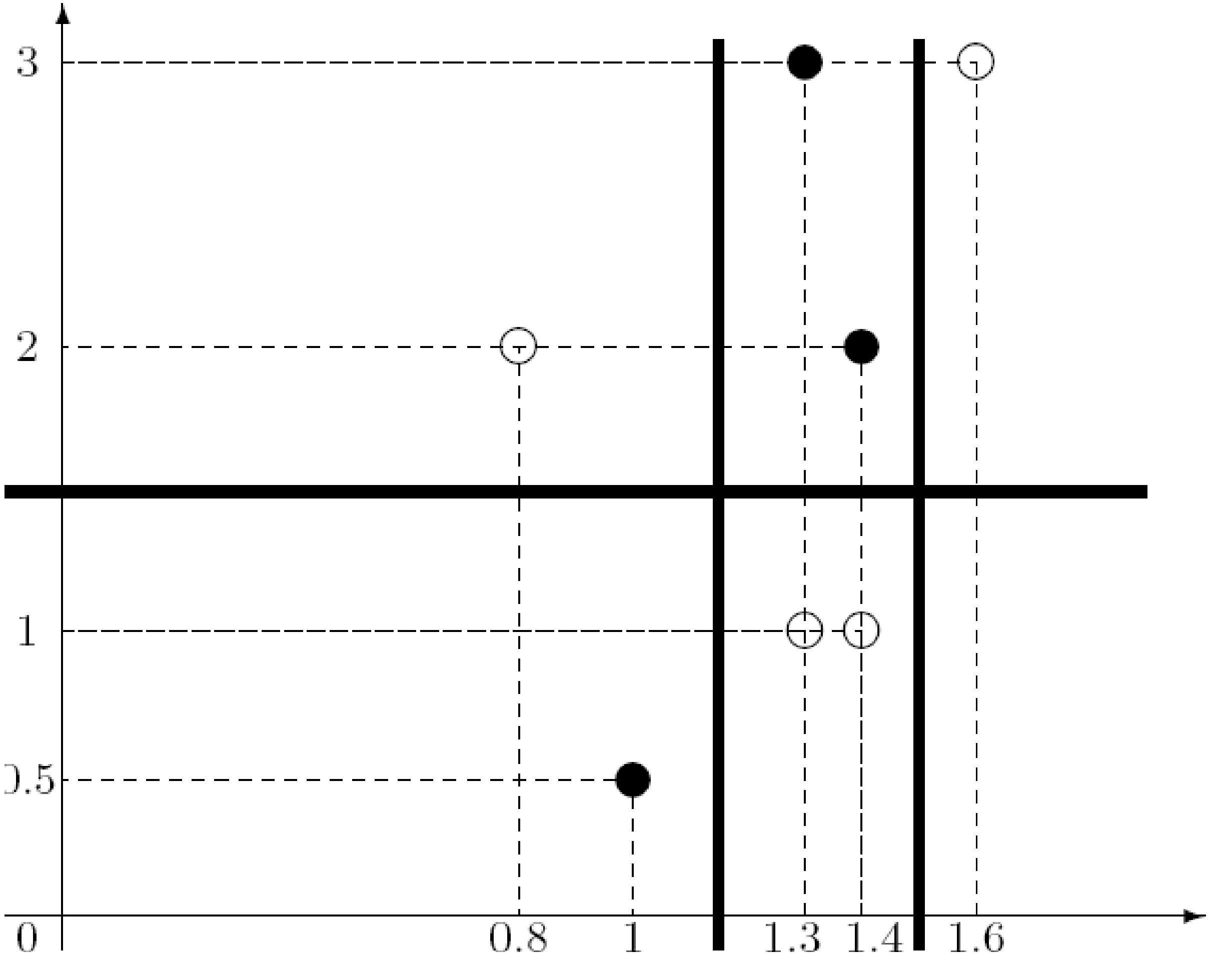
- $p_1^a \sim [0.8; 1)$;
- $p_2^a \sim [1; 1.3)$;
- $p_3^a \sim [1.3; 1.4)$;
- $p_4^a \sim [1.4; 1.6)$;

b:

- $p_1^b \sim [0.5; 1)$;
- $p_2^b \sim [1; 2)$;
- $p_3^b \sim [2; 3)$;

Example: Encoding function

$$\psi_{2,1} = p_1^a + p_1^b + p_2^b; \qquad \psi_{2,4} = p_2^a + p_3^a + p_1^b;$$
$$\psi_{2,6} = p_2^a + p_3^a + p_4^a + p_1^b + p_2^b + p_3^b; \qquad \psi_{2,7} = p_2^a + p_1^b;$$
$$\psi_{3,1} = p_1^a + p_2^a + p_3^b; \qquad \psi_{3,4} = p_2^a + p_2^b + p_3^b;$$
$$\psi_{3,6} = p_3^a + p_4^a; \qquad \psi_{3,7} = p_2^b + p_3^b;$$
$$\psi_{5,1} = p_1^a + p_2^a + p_3^a; \qquad \psi_{5,4} = p_2^b;$$
$$\psi_{5,6} = p_4^a + p_3^b; \qquad \psi_{5,7} = p_3^a + p_2^b;$$



$$\Phi_{\mathbb{S}} = \left(p_1^a + p_1^b + p_2^b\right)\left(p_1^a + p_2^a + p_3^b\right)\left(p_1^a + p_2^a + p_3^a\right)\left(p_2^a + p_3^a + p_1^b\right) p_2^b \left(p_2^a + p_2^b + p_3^b\right)$$
$$\left(p_2^a + p_3^a + p_4^a + p_1^b + p_2^b + p_3^b\right)\left(p_3^a + p_4^a\right)\left(p_4^a + p_3^b\right)\left(p_2^a + p_1^b\right)\left(p_2^b + p_3^b\right)\left(p_3^a + p_2^b\right)$$

$$\Phi_{\mathbb{S}} = p_2^a p_4^a p_2^b + p_2^a p_3^a p_2^b p_3^b + p_3^a p_1^b p_2^b p_3^b + p_1^a p_4^a p_1^b p_2^b.$$

MD-heuristics

- A supervised, dynamic discretization method
- Quality of a cut = number of pairs discerned by this cut
- Both local and global versions are possible
- Global version may have high time complexity ($O(n^3k)$ per cut)
- Time complexity can be reduced by using additional data structure ($O(nk \log n)$ per cut)

Algorithm 2 MD-heuristic for optimal discretization problem

Require: Decision table $\mathbb{S} = (U, A, dec)$

Ensure: The semi-optimal set of cuts;

1: *Construct the table* $\mathbb{S}^*$ *from* $\mathbb{S}$ *and set* $\mathbf{B} := \mathbb{S}^*$;
2: *Select the column of* $\mathbf{B}$ *with the maximal number of occurrences of 1's;*
3: *Delete from* $\mathbf{B}$ *the selected column in Step 2 together with all rows marked in this column by 1;*
4: **if** $\mathbf{B}$ *consists of more than one row* **then**
5:     go to Step 2
6: **else**
7:     Return the set of selected cuts as a result;
8:     Stop;
9: **end if**

MD heuristics

$\mathbb{S}^*$	p_1^a	p_2^a	p_3^a	p_4^a	p_1^b	p_2^b	p_3^b	d^*
(u_1, u_2)	1	0	0	0	1	1	0	1
(u_1, u_3)	1	1	0	0	0	0	1	1
(u_1, u_5)	1	1	1	0	0	0	0	1
(u_4, u_2)	0	1	1	0	1	0	0	1
(u_4, u_3)	0	0	1	0	0	1	1	1
(u_4, u_5)	0	0	0	0	0	1	0	1
(u_6, u_2)	0	1	1	1	1	1	1	1
(u_6, u_3)	0	0	1	1	0	0	0	1
(u_6, u_5)	0	0	0	1	0	0	1	1
(u_7, u_2)	0	1	0	0	1	0	0	1
(u_7, u_3)	0	0	0	0	0	1	1	1
(u_7, u_5)	0	0	1	0	0	1	0	1
new	0	0	0	0	0	0	0	0

Improved algorithm

- DTree - a modified decision tree structure for discretization.
- Possible operations:
  - Init(S): initializes the data structure for the given decision table;
  - Conflict(): returns the number of pairs of undiscerned objects;
  - GetBestCut(): returns the best cut point with respect to the discernibility measure;
  - InsertCut(a, c): inserts the cut (a, c) and updates the data structure.
- Init(S) requires O(nk log n)
- The rest requires O(nk) only.

Improved algorithm

Algorithm 3 Implementation of **MD-heuristic** using *DTree* structure

Require: Decision table $\mathbb{S} = (U, A, dec)$
Ensure: The semi-optimal set of cuts;

```
DTree D = new DTree();
D.Init(S);
while (D.Conflict() > 0) do
    Cut c = D.GetBestCut();
    if (c.quality== 0) then
        break;
    end if
    D.InsertCut(c.attribute,c.cutpoint);
end while
endwhile
D.PrintCuts();
```

Properties of MD-heuristics

- Boundary cuts
- Discretization problem in $\mathbf{R}^2$ still remains NP-hard
- Local MD-heuristics for discretization → decision tree
- Attribute reduction vs. discretization