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, „kdnugget”, 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 $S = (U, A \setminus \{d\})$ 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 $C$. 

\begin{figure}[h]
\centering
\includegraphics[width=\columnwidth]{diagram.png}
\caption{Discernibility by cuts diagram.}
\end{figure}
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 $\text{dec}(x) \neq \text{dec}(y)$, the following condition holds:

**IF** $x, y$ are discernible by $A$ 
**THEN** $x, y$ are discernible by $C$
Optimal discretization problem

\textbf{OPTI\textsc{Disc}}: optimal discretization problem
\begin{itemize}
  \item \textit{input}: A decision table $S$.
  \item \textit{output}: $S$-optimal set of cuts.
\end{itemize}

\textbf{Disc\textsc{Size}}: $k$-cuts discretization problem
\begin{itemize}
  \item \textit{input}: A decision table $S$ and an integer $k$.
  \item \textit{question}: Decide whether there exists a $S$-irreducible set of cuts $P$ such that $\text{card}(P) < k$.
\end{itemize}

\textbf{Theorem 2 (Computational complexity of discretization problems)}.

1. Disc\textsc{Size} is $NP$-complete.
2. Opti\textsc{Disc} 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 \((a, c) \in C\) a Boolean variable \(p_{(a,c)}\)

- \(p_{(a,c)} = 1 \iff \) the cut \((a, c)\) is selected
Encoding function

- For any pair of objects $u_i, u_j \in U$.
  \[ X_{i,j}^a = \{ (a, c_k^a) \in C_a : (a(u_i) - c_k^a)(a(u_j) - c_k^a) < 0 \} \cdot \]
  \[ X_{i,j} = \bigcup_{a \in A} X_{i,j}^a \]

- Discernibility function for two objects
  \[ \psi_{i,j} = \begin{cases} 
  \Sigma X_{i,j} & \text{if } X_{i,j} \neq \emptyset \\
  1 & \text{if } X_{i,j} = \emptyset 
  \end{cases} \]

- Discernibility function for discretization problem
  \[ \Phi_S = \prod_{d(u_i) \neq d(u_j)} \psi_{i,j}. \]
Example: Boolean variables

<table>
<thead>
<tr>
<th>S</th>
<th>a</th>
<th>b</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>0.8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$u_2$</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>$u_3$</td>
<td>1.3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>$u_4$</td>
<td>1.4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$u_5$</td>
<td>1.4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$u_6$</td>
<td>1.6</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$u_7$</td>
<td>1.3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

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

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

$b$:  
- $p_1^b \sim [0.5; 1]$;
- $p_2^b \sim [1; 2]$;
- $p_3^b \sim [2; 3]$;
Example: Encoding function

\[ \psi_{2,1} = p_1^a + p_1^b + p_2^b; \]
\[ \psi_{2,6} = p_2^a + p_3^a + p_4^a + p_1^b + p_2^b + p_3^b; \]
\[ \psi_{3,1} = p_1^a + p_2^a + p_3^b; \]
\[ \psi_{3,6} = p_3^a + p_4^a; \]
\[ \psi_{5,1} = p_1^a + p_2^a + p_3^a; \]
\[ \psi_{5,6} = p_4^a + p_3^b; \]
\[ \psi_{2,4} = p_2^a + p_3^a + p_1^b; \]
\[ \psi_{2,7} = p_2^a + p_1^b; \]
\[ \psi_{3,4} = p_2^a + p_2^b + p_3^b; \]
\[ \psi_{3,7} = p_2^b + p_3^b; \]
\[ \psi_{5,4} = p_2^b; \]
\[ \psi_{5,7} = p_3^a + p_2^b; \]
\[ \Phi_S = \left( p_1^a + p_1^b + p_2^b \right) \left( p_1^a + p_2^a + p_3^b \right) \left( p_1^a + p_2^a + p_3^a \right) \left( p_2^a + p_3^a + p_1^b \right) p_2^b \left( p_2^a + p_2^b + p_3^b \right) \]

\[ \Phi_S = p_2^a p_4^a p_2^b + p_2^a p_3^a p_2^b p_3^b + p_3^a p_4^b p_2^b p_3^b + p_1^a p_4^a p_1^b p_2^b. \]
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)
\textbf{Algorithm 2} MD-heuristic for optimal discretization problem

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

\textbf{Ensure:} The semi-optimal set of cuts;

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

<table>
<thead>
<tr>
<th>( S^* )</th>
<th>( p_1^a )</th>
<th>( p_2^a )</th>
<th>( p_3^a )</th>
<th>( p_4^a )</th>
<th>( p_1^b )</th>
<th>( p_2^b )</th>
<th>( p_3^b )</th>
<th>( d^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>((u_1, u_2))</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>((u_1, u_3))</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>((u_1, u_5))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>((u_4, u_2))</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>((u_4, u_3))</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>((u_4, u_5))</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>((u_6, u_2))</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>((u_6, u_3))</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>((u_6, u_5))</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>((u_7, u_2))</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>((u_7, u_3))</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>((u_7, u_5))</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>new</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
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 $S = (U, A, dec)$
Ensure: The semi-optimal set of cuts;

1: $DTree D = \text{new } DTREE();$
2: $D.$Init($S$);
3: while ($D.$Conflict() > 0) do
4:     $Cut c = D.$GetBestCut();
5:     if ($c.$quality == 0) then
6:         break;
7:     end if
8:     $D.$InsertCut($c.$attribute, $c.$cutpoint);
9: end while
10: endwhile
11: $D.$PrintCuts();
Properties of MD-heuristics

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