Post-processing of Decision Trees

• Three methods developed for these objectives
• Based on machine learning and supervised learning
• Under the evolutionary paradigm
  – specifically Genetic Programming (GP)

Objectives:
1. Improve precision and recall
2. Classify minority class in extreme imbalanced datasets
2. Produce a range of rules to suit user's preferences
3. Generate comprehensible solutions for user interaction.
New Classification Methods

Scenario Method

Repository Method

Evolving Decision Rules = Repository Method + Evolution

(Pruning)

(Rules Collection)
### Methods overview

| Repository Method (RM) | 1. Classify minority class in imbalanced data sets  
2. Produce a range of classifications to suit the user's preferences  
3. Provide understandable rules | Our aim is to extract and collect different patterns that classify the positive cases (rare instances) in different ways. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolving decision rules (EDR)</td>
<td></td>
</tr>
</tbody>
</table>
EDR evolves a population of decision trees to form a repository of rules. The resulting rules are used to create a range of classifications |
| Scenario Method (SM) | 1. Analyze decision tree to detect and remove the rules that do not contribute to the classification task. | Scenario Method (SM) is a pruning procedure for decision trees created by GP. This pruning is based on the analysis of patterns in the decision tree. |
Repository Method

In order to mine the knowledge acquired by the evolutionary process, Repository Method performs the following steps:

1- Rule extraction
   Evolve a GP to create a population of decision trees

2- Rule simplification
   \[
   \begin{align*}
   &R_1 \quad \text{The rule } R_k \text{ is selected by } \textbf{precision}; \\
   &R_2 \\
   &\ldots \\
   &R_n \\
   &R_k \text{ is simplified to } R'_k
   \end{align*}
   \]

3- New rule detection
   \[R'_k \text{ is compared to the rules in the repository by similarity (genotype)}\]

4- Add rule to the repository
   If \(R'_k\) is a new rule, \(R'_k\) is added to the rule repository
Evolving Decision Trees

The new population is processed until the maximum number of iterations is reached.

- **Initial Population generated at random**
- Every decision tree is divided into rules (patterns)

  \{ R_1, R_2, \ldots, R_3 \}

- **Every Rule**

  - Every Rule $R_k$ whose precision achieve a predefined precision threshold is simplified to remove redundant and vacuous conditions.

- **$R'_{k}$**

  - Is compared to the rules in a rule collection.
  - If $R'_{k}$ is a new rule, it is added to the rule set.

- **generate a new population of decision tree by mutating and hill-climbing on the rules in the repository and generating trees at random**

EDR Presentation
Scenario Method

The tree is composed by the following rules:

- **R1 = [2, 7, 10]**
- **R2 = [2, 14, 18]**
- **R3 = [2, 14, 21]**

Where the numbers represent the node of the condition.

**Procedure**

- Evaluate every decision rule
- Consider that the R2 is not contributing to the classification task. Thus, we analyze every condition in R2 to determine which conditions are involved in other rules.
- The only condition that is not involved in the other rules is 18.
- Remove the condition.