Rule Compressor™

Using Machine Learning for Compression of Large Classification Rulesets

Jacob Feldman, PhD
Chief Technology Officer

Business Rules (BR)

- Business Rules and Decision Management Systems are commonly used to represent, manage, and execute business logic efficiently using Rule Engines

Machine Learning (ML)

- Machine Learning offers powerful algorithms and tools for practical extraction of rules (patterns) from data
Real-World BR&ML Problems

- It becomes increasingly important to find previously unknown dependencies inside data streams
- Online multi-transactional processing systems require new rules to be discovered “on-the-fly”
- Business Rules Repositories grow quickly, become too complicated, and have to be compressed and optimized
ML + BR Integration

- Rule Learner discovers and produces rules
- Rule Engine consumes/executes them
OpenRules Rule Learner: ML+BR Integration Schema

- **Enterprise Data Repository**
  - Labeled Data Instances
  - Test Data Instances
  - Real-Time Data
  - Historical Data

- **Rule Trainer**
  - "Train Rule Learner"

- **Rule Learner**
  - "Discover Rules"

- **Rule Converter**
  - "Convert Rules"

- **Rule Engine**
  - "Execute Rules"

- **Business Rules Repository**
  - Training Rules
  - Learned Rules (ML output format)
  - Learned Rules (Human Readable & Rule Engine Processable)
OpenRules Rule Learner: ML+BR Integration Schema

- **Enterprise Data Repository**
  - Labeled Data Instances
  - Test Data Instances
  - Real-Time Data
  - Historical Data

- **Business Rules Repository**
  - Training Rules
  - Learned Rules (ML output format)
  - Learned Rules (Human Readable & Rule Engine Processable)

**Ever-Learning Loop**

**ML**

**BR**

**Discover Rules**

**Train Rule Learner**

**Rule Learner**

**Rule Trainer**

**Rule Engine**

"Execute Rules"

**Rule Converter**

"Convert Rules"
“Why not build machine learners that learn in the same cumulative way as humans, becoming increasingly competent rather than halting at some plateau? Can we build never-ending learners?”

Tom M. Mitchell, Carnegie Mellon University
**Real-World Examples**

- In a real-world application, humans experts classify their data using a “gut feel” based on their past experience in working with the data.

- Rule Learner helps to convert this “gut feel” into rules with very specific numeric thresholds!

- Example of generated red-flag rules for a large government agency:

  ```
  Rules String classifyCarExpense(Record r) | Rule 1 | Rule 2 | Rule 3
  if CAR_EXPENSE_AMOUNT >= 2758 |         |        |
  and BUSINESS_MILES_COUNT >= 4100 |        |        |
  and GROSS_RECEIPTS_AMOUNT <= 3772 |        |        |
  then CAR_EXPENSE | RED     | RED    | GREEN
  ```

  - Rule Learner selected only a few essential attributes out of hundreds considering around 50K data instances
Motivation for BR Compression

- Decision Tables and other rulesets have a tendency to grow quickly
  - Attempts to cover all possible combinations of decision variables lead to a combinatorial explosion
  - A decision table with 5-10 columns may end up with 1000’s rules

- Question: Can ML help with compression of large rulesets?
Manual Rules Compression

18 rules => 6 rules
(with overrides)

One cannot do it manually when there are much more attributes and rules – we need an automatic compression!
Automatic Rules Compression

### Correctly Classified Instances:
2,395 out of 2,396 instances were correctly classified.

### Example Rules Compression

<table>
<thead>
<tr>
<th>IF type is</th>
<th>AND adjustment &gt; $500</th>
<th>AND adjustment &lt; $500</th>
<th>AND amount &lt; $500</th>
<th>AND amount &gt;= $500</th>
<th>THEN Classify Instance as</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>$200</td>
<td></td>
<td>-$150</td>
<td></td>
<td>TOP</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td>$200</td>
<td>-$189</td>
<td></td>
<td>BOTTOM</td>
</tr>
<tr>
<td>32</td>
<td>$500</td>
<td></td>
<td>-$1,000</td>
<td></td>
<td>TOP</td>
</tr>
<tr>
<td>32</td>
<td>$500</td>
<td></td>
<td>-$99</td>
<td></td>
<td>BOTTOM</td>
</tr>
<tr>
<td>33</td>
<td>$500</td>
<td></td>
<td>-$1,000</td>
<td></td>
<td>TOP</td>
</tr>
<tr>
<td>33</td>
<td></td>
<td>$500</td>
<td>-$100</td>
<td></td>
<td>BOTTOM</td>
</tr>
<tr>
<td>34</td>
<td>$500</td>
<td></td>
<td>-$1,000</td>
<td></td>
<td>TOP</td>
</tr>
<tr>
<td>34</td>
<td></td>
<td>$500</td>
<td>-$100</td>
<td></td>
<td>BOTTOM</td>
</tr>
<tr>
<td>35</td>
<td>$500</td>
<td></td>
<td>-$800</td>
<td></td>
<td>TOP</td>
</tr>
<tr>
<td>35</td>
<td>$500</td>
<td></td>
<td>-$100</td>
<td></td>
<td>BOTTOM</td>
</tr>
<tr>
<td>36</td>
<td>$500</td>
<td></td>
<td>-$800</td>
<td></td>
<td>TOP</td>
</tr>
<tr>
<td>36</td>
<td></td>
<td>$500</td>
<td>-$100</td>
<td></td>
<td>BOTTOM</td>
</tr>
<tr>
<td>37</td>
<td>$500</td>
<td></td>
<td>-$2,000</td>
<td></td>
<td>TOP</td>
</tr>
<tr>
<td>37</td>
<td></td>
<td>$500</td>
<td>$0.0</td>
<td></td>
<td>BOTTOM</td>
</tr>
</tbody>
</table>

15 rules => 1 rule!
How Rule Compressor Works

1. Test Data
   - Execute Manual Rules to classify test data

2. Training Set
   - Convert Test Data to Training Set

3. Compressed rules
   - Execute Rule Learner
   - Execute Rule Converter

4. Excel rules
   - Execute OpenRules Engine and compare the results

5. Comparison results
Automatic Rules Generation: Important Warning

- **Positive effect:**
  - Smaller and easy to maintain rulesets

- **Negative effect:**
  - Unavoidable errors: always validate if generated rules are acceptable
    - Could be OK for small insurance claims
    - Could be a disaster for medical diagnostics
  - Lost business knowledge
Conclusion

- ML+BR integration brings immediate improvements to BR systems by supporting never-ending rules discovery and adjustment.
- Rule Compressor allows compressing large rules sets.
- It is always necessary to evaluate if unavoidable errors are acceptable.