Incremental Approach to Interpretable Classification Rule Learning
Bishwamittra Ghosh and Kuldeep S. Meel
School of Computing, National University of Singapore

Interpretable Machine Learning
- The wide adoption of machine learning in the critical domains has propelled the need for interpretable techniques
- Interpretable machine learning model provides end users the reasoning behind decision-making
- We propose an incremental approach for learning interpretable classification rules

Examples of Interpretable Rules
- Credit Default Dataset
  A client payment defaults if
  Repayment status in September: payment delay > 1 month OR
  Repayment status in August: payment delay > 2 months OR
  Repayment status in June: payment delay > 2 months OR
  Education type = other
- Pima Indian Diabetes Dataset
  A person is tested positive for diabetes if
  Plasma glucose concentration > 125 AND
  Triceps skin fold thickness ≤ 35 mm AND
  Diabetes pedigree function > 0.259 AND
  Age > 25 years

Learning Interpretable Classification Rules
- In rule-based classifiers, sparsity refers to the interpretability of the rule, i.e., a sparser rule is more interpretable [1]
- Consider decision variables:
  \[-b_i^j = I\{j-th feature is selected in i-th clause\}\]
  \[-\eta_q = I\{sample q is misclassified\}\]
- To learn an interpretable classification rule, the objective function is:
  \[
  \min \sum_{i,j} b_i^j + \lambda \sum_q \eta_q
  \]
- Constraints: positive labeled samples satisfy the rule, and negative labeled samples do not satisfy the rule
- In MaxSAT, the objective function is encoded as soft clauses and the constraints are encoded as hard clauses

Analysis
- To generate a k-clause CNF rule for a dataset of n samples over m boolean features, the number of clauses of the MaxSAT instance is \(O(n \cdot m \cdot k)\)
- Suffers from poor scalability when dataset is large

References

An Incremental Rule-learning Approach
- We attribute large formula size of the MaxSAT instance for the poor scalability
- We propose a mini-batch incremental learning framework with the following objective function
  \[
  \min \sum_{i,j} b_i^j \cdot I(b_i^j) + \lambda \sum_q \eta_q.
  \]
  where indicator function \(I(\cdot)\) is defined as follows.
  \[
  I(b_i^j) = \begin{cases} 
  -1 & \text{if } b_i^j = 1 \text{ in the } (t-1)\text{-th batch } (t \neq 1) \\
  1 & \text{otherwise}
  \end{cases}
  \]

Solution Technique
- Divide the training data into a fixed number of batches \(p\)
- The MaxSAT instance constructed for the \(t\)-th batch depends on the training data in the \((t-1)\)-th batch and the rule learned in the \((t-1)\)-th batch
- Construct soft unit clauses to encode the learned assignment of decision variables in the previous batch

Key Contribution
IMLI makes \(p\) queries to the MaxSAT solver with each query of the formula size \(O(\frac{2}{p} \cdot m \cdot k)\)

Experimental Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size n</th>
<th>Features m</th>
<th>LR (s)</th>
<th>SVC (s)</th>
<th>RIPPER (s)</th>
<th>IMLI (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIMA</td>
<td>768</td>
<td>134</td>
<td>75.32</td>
<td>75.32</td>
<td>75.32</td>
<td>75.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.3)</td>
<td>(0.37)</td>
<td>(2.58)</td>
<td>(0.74)</td>
</tr>
<tr>
<td>Credit-default</td>
<td>30000</td>
<td>334</td>
<td>80.81</td>
<td>80.69</td>
<td>80.97</td>
<td>79.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(6.67)</td>
<td>(6.87)</td>
<td>(8.47)</td>
<td>(32.58)</td>
</tr>
<tr>
<td>Twitter</td>
<td>49999</td>
<td>1050</td>
<td>95.67</td>
<td>95.56</td>
<td>95.56</td>
<td>94.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(3.99)</td>
<td>(98.21)</td>
<td>(98.21)</td>
<td>(59.67)</td>
</tr>
</tbody>
</table>

Table 1: Each cell in the last 5 columns refers to test accuracy (%) and training time (s). IMLI exhibits better training time by costing a little bit of accuracy.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RIPPER</th>
<th>IMLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIMA</td>
<td>8.25</td>
<td>3.5</td>
</tr>
<tr>
<td>Twitter</td>
<td>21.6</td>
<td>6</td>
</tr>
<tr>
<td>Credit</td>
<td>14.25</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2: Average size of the rules of different rule-based models. IMLI generates shorter rules compared to other rule-based models.

Conclusion
- IMLI achieves up to three orders of magnitude improvement in training time by sacrificing a bit of accuracy
- The generated rules appear to be more interpretable