Multi-objective learning of hybrid classifiers

Rok Piltaver, Mitja Luštrek, Jernej Zupančič, Sašo Džeroski, Matjaž Gams
Jožef Stefan Institute (Ljubljana, Slovenia)

ECAI 2014, August 18-22, Prague
Outline

• Motivation
• Multi-objective learning
• The algorithm
  – Basic algorithm idea
  – Formal problem definition
  – The naive algorithm
  – The efficient algorithm
• Evaluation
  – Naive vs. efficient algorithm
  – Improvements over baseline
  – Comparison with NSGA-II (multi-objective optimization algorithm)
  – Expert validation
• Conclusion
Motivation
Motivation

- **Comprehensible** classifiers (trees, rules, etc.) enable:
  - classification explanation
  - classifier validation
  - knowledge discovery
  - classifier improvement
- Incomprehensible *(black-box)* classifiers (SVM, ANN, ensemble, etc.) often achieve higher **accuracy**.

- Domain experts can often explain **only some** of their **knowledge** with **comprehensible** rules.
- Use classifiers that enable trading accuracy for comprehensibility and vice versa.
Hybrid trees

Classify simple parts of the domain with classification tree, the rest with an accurate black-box classifier.
Multi-objective learning

vs. the traditional approach
Multi-objective learning

The best classifier according to a single objective

Accuracy
Multi-objective learning

Which classifier is the best according to the two objectives?
Multi-objective learning

- Classifier $C_4$ dominates $C_3$, i.e. $C_3$ is dominated by $C_4$. 
Multi-objective learning

- Classifier $C_4$ dominates $C_3$.
- Classifiers $C_4$ and $C_6$ are incomparable.
Multi-objective learning

- Classifier $C_4$ dominates $C_3$.
- Classifiers $C_4$ and $C_6$ are incomparable.
- Non-dominated classifiers (Pareto set).
- Solution is a set of classifiers because some classifiers are incomparable considering the two objectives.
The algorithm

Basic algorithm idea
Formal problem definition
The naive algorithm
The efficient algorithm
Basic algorithm idea

• **Input**: dataset, classification tree, BB classifier.

```
+ comprehensible - not comprehensible
- lower accuracy + higher accuracy
```

```
     A2
  /\   /\  \\
 /   \ /   \
A3    A1  L4  L5
|   / \   |
L1  A1  L2  L3
    |    |   |
    + comprehensible  - lower accuracy
    - not comprehensible  + higher accuracy

Black-box Classifier
```
Basic algorithm idea

- **Input**: dataset, classification tree, BB classifier.
  - Replace some leaves for BB leaves to produce hybrid trees.
Basic algorithm idea

- **Input**: dataset, classification tree, BB classifier.
  - Replace some leaves for BB leaves to produce hybrid trees.
  - Check all possible combinations of replaced leaves.
  - Return only the non-dominated hybrid trees.
- **Output**: set of non-dominated hybrid trees.

![Diagram of hybrid trees]
Basic algorithm idea

- **Input**: dataset, classification tree, BB classifier.
  - Replace some leaves for BB leaves to produce hybrid trees.
  - Check all possible combinations of replaced leaves.
  - Return only the non-dominated hybrid trees.
- **Output**: set of non-dominated hybrid trees.

\[
\begin{align*}
\text{hybrid trees accuracy-comprehensibility trade-off} \\
0 \leq \frac{\#\text{inst. in reg. leaves}}{\#\text{all instances}} \leq 1
\end{align*}
\]
Multi-objective DM process

provide: dataset, initial tree, BB classifier

repeat

find a set of hybrid trees
expert analyzes the hybrid trees
expert selects the best hybrid tree or
proposes new input parameters

until satisfied
Formal problem definition

Hybrid tree: \( T_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,n}) \in \{0, 1\}^n \)

- Hybrid trees are represented by binary vectors with as many components as there are leaves in the initial tree \( (n) \).
- Value 0 represents a regular leaf.
- Value 1 represents a BB leaf.

\[ T_i = (0, 1, 0, 0, 1, 1, 0) \]
Formal problem definition

Hybrid tree: $T_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,n}) \in \{0, 1\}^n$

Search space: $\{0, 1\}^n = \{0, 1\} \times \{0, 1\} \times \ldots \times \{0, 1\}$

Search through the space of $2^n$ hybrid trees.
Formal problem definition

Hybrid tree: $T_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,n}) \in \{0, 1\}^n$
Search space: $\{0, 1\}^n = \{0, 1\} \times \{0, 1\} \times \ldots \times \{0, 1\}$

Quality of hybrid tree $T_i = (a_i, c_i) \in [0, 1] \times [0, 1]$
- $a_i = (\sum_{j \in \text{regular leaf}} N_{j,t} + \sum_{j \in \text{hybrid leaf}} N_{j,bb}) / N$
- $c_i = (\sum_{j \in \text{regular leaf}} N_j) / N$

Evaluate the quality of each hybrid trees: $O(n)$ operations needed in each of $2^n$ hybrid trees.
Dynamic programming approach decreases $O(n2^n)$ to $O(2^n)$:

- $t_{j,k} = t_{i,k}$ if $k \neq l$ else $t_{j,l} = 1$
- $T_j = (t_{i,1}, t_{i,2}, \ldots, 1, \ldots, t_{i,n})$
- $\delta_{l,a} = (N_{l,bb} - N_{l,t})/N > 0$
- $\delta_{l,c} = -N_l/N < 0$
- $(a_j, c_j) = (a_i + \delta_{l,a}, c_i + \delta_{l,c})$

Replace a single leaf for BB, add constants instead of summation.
Replacing a leaf for BB

Split the data into subsets belonging to each leaf and count the number of correctly classified instances.
Replacing a leaf for BB

<table>
<thead>
<tr>
<th>leaf $i$</th>
<th>$1$</th>
<th>$2$</th>
<th>$3$</th>
<th>$4$</th>
<th>$\Sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>#inst. In leaf $N_i$</td>
<td>$8$</td>
<td>$9$</td>
<td>$7$</td>
<td>$10$</td>
<td>$34$</td>
</tr>
<tr>
<td>#correctly classified using the tree $N_{i,t}$</td>
<td>$4$</td>
<td>$5$</td>
<td>$4$</td>
<td>$6$</td>
<td>$19$</td>
</tr>
<tr>
<td>#correctly classified using the BB $N_{i,\text{BB}}$</td>
<td>$8$</td>
<td>$7$</td>
<td>$7$</td>
<td>$7$</td>
<td>$29$</td>
</tr>
<tr>
<td>dif. in accuracy $\delta_{i,a}$</td>
<td>$-4$</td>
<td>$+2$</td>
<td>$+3$</td>
<td>$+1$</td>
<td>$10$</td>
</tr>
<tr>
<td>dif. in compr. $\delta_{i,c}$</td>
<td>$8$</td>
<td>$-9$</td>
<td>$-7$</td>
<td>$-10$</td>
<td>$34$</td>
</tr>
</tbody>
</table>
Replacing a leaf for BB

<table>
<thead>
<tr>
<th>leaf $i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>#inst. In leaf $N_i$</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>10</td>
<td>34</td>
</tr>
<tr>
<td>#correctly classified using the tree $N_{i,t}$</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>#correctly classified using the BB $N_{i,BB}$</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>29</td>
</tr>
<tr>
<td>dif. in accuracy $δ_{i,a}$</td>
<td>-4</td>
<td>+2</td>
<td>+3</td>
<td>+1</td>
<td>10</td>
</tr>
<tr>
<td>dif. in compr. $δ_{i,c}$</td>
<td>8</td>
<td>-9</td>
<td>-7</td>
<td>-10</td>
<td>34</td>
</tr>
</tbody>
</table>
Replacing a leaf for BB

Hybrid tree with 1 BB leaf

<table>
<thead>
<tr>
<th>leaf $i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>∑</th>
</tr>
</thead>
<tbody>
<tr>
<td>#inst. In leaf $N_i$</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>10</td>
<td>34</td>
</tr>
<tr>
<td>#correctly classified using the tree $N_{i,t}$</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>#correctly classified using the BB $N_{i,BB}$</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>29</td>
</tr>
<tr>
<td>dif. in accuracy $\delta_{i,a}$</td>
<td>+4</td>
<td>+2</td>
<td>+3</td>
<td>+1</td>
<td>10</td>
</tr>
<tr>
<td>dif. in compr. $\delta_{i,c}$</td>
<td>8</td>
<td>-9</td>
<td>-7</td>
<td>-10</td>
<td>34</td>
</tr>
</tbody>
</table>
Replacing a leaf for BB

<table>
<thead>
<tr>
<th>leaf $i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>$\Sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>#inst. In leaf $N_i$</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>10</td>
<td>34</td>
</tr>
<tr>
<td>#correctly classified using the tree $N_{i,t}$</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>#correctly classified using the BB $N_{i,BB}$</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>29</td>
</tr>
<tr>
<td>dif. in accuracy $\delta_{i,a}$</td>
<td>+4</td>
<td>+2</td>
<td>+3</td>
<td>+1</td>
<td>10</td>
</tr>
<tr>
<td>dif. in compr. $\delta_{i,c}$</td>
<td>8</td>
<td>-9</td>
<td>-7</td>
<td>-10</td>
<td>34</td>
</tr>
</tbody>
</table>
The naive algorithm

- Start with the initial tree.
The naive algorithm

- Replace one leaf to obtain 4 hybrid trees with one BB leaf and three regular leaves.
The naive algorithm

- Replace any of the remaining regular leaves in the obtained hybrid tree.
The naive algorithm

- Obtain a set of leaves with 0, 1 or 2 BB leaves.
The naive algorithm

- Repeat replacing leaves.
The naive algorithm

- Until all the leaves are replaced, i.e. hybrid tree becomes the BB classifier.
The naive algorithm

- Select only the non-dominated hybrid trees.
Some hybrid trees are generated more than once.
Most generated hybrid trees are dominated.
The efficient algorithm

iteration: 1

- Start with the initial tree.
The efficient algorithm

- Select non-dominated leaf according to relative dif. in acc. $\delta_{i,a}$ and compr. $\delta_{i,c}$.
The efficient algorithm

- Replace only the non-dominated leaves to prune the search space.
The efficient algorithm

- Replace only the non-dominated remaining regular leaves in each hybrid tree.

Do not replace leaves with index smaller than the greatest index of replaced leaf.
The efficient algorithm

iteration: 1

\( \delta_{i,c} \) \( \delta_{i,a} \)

\( t_0 \)

• Repeat.
The efficient algorithm

iteration: 1

\[ \delta_{i,c}, \delta_{i,a} \]

\[ t_0 \]

\[ t_{2^n} \]

BB

comprehensibility

accuracy
The efficient algorithm

- Select only the non-dominated hybrid trees.
We have proven:

- Replacing leaves implicitly considers replacing subtrees.
- $\delta_{l,a}$ and $\delta_{l,c}$ do not depend on the state of the tree.
- Efficient algorithm finds the complete Pareto optimal set.
- Efficient algorithm might generate some dominated trees.
Evaluation

Naive vs. efficient algorithm
Improvements over baseline
Comparison with NSGA-II (multi-objective optimization algorithm)
Expert validation
Naive vs. efficient algorithm

Graphs show accuracy and comprehensibility of hybrid trees:

- **red** dots = trees generated by the efficient algorithm,
- **blue** dots = avoided hybrid trees = pruned search space.
Naive vs. efficient algorithm

Naive:
- 17 leaves, 3.26 min
- 18 leaves, 18.08 min
- 20+ leaves, too long

Efficient:
- 17 leaves, 0.48 ms
- 40 leaves, ~0.5 s
- 50+ leaves – not comprehensible to begin with
Improvements over baseline

Baseline solution set: initial classification tree & BB classifier

Comparison measure: hyper-volume (area under Pareto front)
Improvements over baseline

**Baseline solution set:** initial classification tree & BB classifier

**Comparison measure:** hyper-volume (area under Pareto front)

Small difference in hyper volume between the non-dominated set of hybrid trees and the baseline would mean that hybrid trees are not much better than the baseline.
Improvements over baseline

**Baseline solution set:** initial classification tree & BB classifier

**Comparison measure:** hyper-volume (area under Pareto front)

On 23 tested datasets the difference in hyper volume is considerable therefore hybrid trees really provide previously unreachable trade-off between accuracy and comprehensibility.
Improvements over baseline

Baseline solution set: initial classification tree & BB classifier
Comparison measure: hyper-volume (area under Pareto front)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Hybrid tree hypervol.</th>
<th>Baseline hypervol.</th>
<th>Hypervol. difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>mfeat-pixel</td>
<td>1,138</td>
<td>0,833</td>
<td>0,305</td>
</tr>
<tr>
<td>letter</td>
<td>1,108</td>
<td>0,815</td>
<td>0,293</td>
</tr>
<tr>
<td>vowel</td>
<td>1,003</td>
<td>0,772</td>
<td>0,231</td>
</tr>
<tr>
<td>mfeat-zernike</td>
<td>1,154</td>
<td>1,031</td>
<td>0,123</td>
</tr>
<tr>
<td>balance-scale</td>
<td>1,229</td>
<td>1,201</td>
<td>0,029</td>
</tr>
<tr>
<td>cylinder-bands</td>
<td>1,028</td>
<td>1,002</td>
<td>0,026</td>
</tr>
<tr>
<td>flags</td>
<td>0,914</td>
<td>0,896</td>
<td>0,017</td>
</tr>
</tbody>
</table>
A state-of-the-art multi-objective optimization algorithm:
• search limited by runtime (or number of generations),
• stochastic, requires parameter settings.

The proposed algorithm outperforms NSGA-II in terms of hyper-volume because it is exact (NSGA-II is not).

<table>
<thead>
<tr>
<th>Population size/Run-time</th>
<th>Small tree (12.8 leaves)</th>
<th>Big tree (22 leaves)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>big</td>
<td>med</td>
</tr>
<tr>
<td>t × 10</td>
<td>3.8</td>
<td>1.8</td>
</tr>
<tr>
<td>t × 50</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>t × 100</td>
<td>0.02</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Relative difference in hyper-volume (%)

- $t = \text{run-time of the proposed algorithm}$
- NSGA-II run-time $\in \{t \times 10, t \times 50, t \times 100\}$
- NSGA-II population size:
  - med = # hybrid-trees in the Pareto set
  - big = med $\times 2$
  - small = med / 2
- $|\text{search space}| = 2^{\text{tree size}}$
Comparison with NSGA-II

For big search space or short run-time the difference decreases with decreasing population size (increasing number of generations) – NSGA-II does not have enough time (generations) to converge close to the Pareto set, hence the difference in hyper-volume.

<table>
<thead>
<tr>
<th>Population size/Run-time</th>
<th>Small tree (12.8 leaves)</th>
<th>Big tree (22 leaves)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>big</td>
<td>med</td>
</tr>
<tr>
<td>t × 10</td>
<td>3.8</td>
<td>1.8</td>
</tr>
<tr>
<td>t × 50</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>t × 100</td>
<td>0.02</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Relative difference in hyper-volume (%)
Comparison with NSGA-II

For small search space and long run time the difference increases with decreasing size of population – NSGA-II converges to the Pareto set well, however it misses some of the non-dominated hybrid trees if the population size is too small or due to its stochastic nature.

<table>
<thead>
<tr>
<th>Population size/Run-time</th>
<th>Small tree (12.8 leaves)</th>
<th>Big tree (22 leaves)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>big</td>
<td>med</td>
</tr>
<tr>
<td>t × 10</td>
<td>3.8</td>
<td>1.8</td>
</tr>
<tr>
<td>t × 50</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>t × 100</td>
<td>0.02</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Relative difference in hyper-volume (%)
Expert validation
use cases and results summary

Activity recognition domain:
• 10 classes: cycling, running, walking, standing, sitting, transitions, etc.
• 61 attributes: 2 s windows of 3-axis accelerometer (on chest) data
• 48,000 instances, 9 persons (1.5 h each)

Baseline classifier accuracy:
• Accurate BB classifier random forest: 0.906 (can not be validated)
• Comprehensible classifier not accurate enough C4.5: 0.761

Results of proposed algorithm application:
• Expert selected a hybrid tree with accuracy 0.841 and comprehensibility 0.721, i.e. traded 6% of acc. for 72% compr.
• Validated simple activity recognition rules, identified activities that are difficult to distinguish and suggested using an additional accelerometer.
Expert validation

selected hybrid tree

- Black nodes represent the BB leaves.
- Numbers represent classification accuracy of the initial tree.
- Pie charts represent class distribution in nodes.
Expert validation

validated rules

Body upright and moving \(\Rightarrow\) running

Two body angles are flat \(\Rightarrow\) lying

Two body angles are flat

\(\Rightarrow\) lying
Expert validation
black-box part

Difficult to classify sitting, kneeling and standing using classification tree, therefore BB classifier should be used. Alternatively using an additional accelerometer would improve accuracy.
Conclusion
• Classifier accuracy and comprehensibility are both important and should be treated equally.
• Hybrid trees combine comprehensible classification trees with an accurate black-box classifier.
• Multi-objective approach should be used to find the set of non-dominated classifiers providing additional insights into domain.
• Exhaustive search algorithm suffers from combinatorial explosion.
• Proposed algorithm is fast enough for reasonably large (i.e. comprehensible) initial classification trees.
• It outperforms baseline non-dominated set of hybrid trees and multi-objective optimization algorithm.
• Domain expert confirmed usability of the algorithm.
Paper contributions

• an efficient exact algorithm that finds the set of optimal hybrid classifiers according to two conflicting objectives: accuracy and comprehensibility,
• multi-objective iterative data mining process,
• using hybrid trees with black-box leaves to obtain wide range of classifiers: from the most accurate to the most comprehensible,
• hybrid tree comprehensibility measure based on the ratio of instances classified with a comprehensible model.

Supported by the ECCAI Travel Award.