Anytime Algorithms for Learning Anycost Classifiers

Saher Esmeir and Shaul Markovitch
Computer Science Dept.
Technion

- Build a good classifier. Take two weeks.
- Build a classifier. When I need it I’ll let you know.
- Build a classifier. It must be extremely fast and return results within 0.001 of a second.
- Build a very fast classifier. The deadline will be provided when calling it.
- Build a classifier with maximal testing costs of 1000$.
- Build a classifier with maximal testing costs of $X$. $X$ will be supplied at time of classification.

Example: Medical Urgency

Example: Defect Identification

5K labeled previous cases

Saher Esmeir & Shaul Markovitch, Anytime Algorithms for Learning of Anytime Classifiers
**Inductive Learning**

![Diagram of Inductive Learning](image)

**Learning Time**

- **5K labeled previous cases**

![Image of learning time](image)

**Testing & Error Costs**

- **<41°**
- **MRI**

![Image of testing & error costs](image)

**Bounded Test Costs**

- **5K labeled previous cases**

![Image of bounded test costs](image)
Cost in Inductive Learning

- Labeling Costs
- Induction Costs
- Acquisition Costs
- Testing Costs
- Misclassification Costs
- Feature extractor $<f_1, f_2, \ldots, f_n>$

Anytime Algorithms

- Allow resources to be traded for quality
- Performance profile
- Two main classes:
  - Contract
  - Interruptible

Different Cost Scenarios

- Contract Learning
- Interruptible Learning
- Pre-Contract Classification
- Contract Classification
- Interruptible Classification
Various Combinations

CLPC: \( L(E,A,M,\rho^l,\rho^c) \rightarrow h(e) \)
ILPC: \( L(E,A,M,f^l(),\rho^c) \rightarrow h(e) \)
CLCC: \( L(E,A,M,\rho^l) \rightarrow h(e,\rho^c) \)
ILCC: \( L(E,A,M,f^l()) \rightarrow h(e,\rho^c) \)
CLIC: \( L(E,A,M,\rho^l) \rightarrow h(e,f^c()) \)
ILIC: \( L(E,A,M,f^l()) \rightarrow h(e,f^c()) \)

Why Decision Trees?

- A key advantage of trees is their interpretability:
  - Validation
  - Explanation
  - Discovery of relationships

- Additional advantages:
  - Accuracy
  - Classification costs
  - Simplicity of use

Anytime Induction of Small, Accurate Trees

Constraint: Learning Costs

- Due to the importance of the tree, we are willing to wait longer if a better one can be obtained

- **Goal:** Design an induction algorithm that can exploit additional resources to induce better trees

- Many real-life applications that involve offline learning can benefit
The Space of Trees

Consistent Trees

What Tree to Prefer?

Occam’s Razor: smaller consistent trees are better

Also for non-consistent!

<table>
<thead>
<tr>
<th>DATASET</th>
<th>ACC.</th>
<th>RTG SIZE</th>
<th>ρ</th>
</tr>
</thead>
<tbody>
<tr>
<td>BREAST-W</td>
<td>92.9 ± 2.8</td>
<td>128 ± 14</td>
<td>-0.1 7</td>
</tr>
<tr>
<td>BUPA</td>
<td>59.7 ± 8.0</td>
<td>213 ± 10</td>
<td>0 10</td>
</tr>
<tr>
<td>CAR</td>
<td>72.8 ± 4.6</td>
<td>647 ± 77</td>
<td>-0.8 10</td>
</tr>
<tr>
<td>CLEVELAND</td>
<td>51.6 ± 9.9</td>
<td>188 ± 7</td>
<td>0 6</td>
</tr>
<tr>
<td>CORRAL</td>
<td>73.3 ± 22.9</td>
<td>15 ± 3</td>
<td>-0.1 9</td>
</tr>
<tr>
<td>GLASS</td>
<td>55.6 ± 9.9</td>
<td>135 ± 8</td>
<td>-0.1 10</td>
</tr>
<tr>
<td>HUNGERIAN</td>
<td>72.8 ± 7.4</td>
<td>125 ± 10</td>
<td>-0.1 9</td>
</tr>
<tr>
<td>IRIS</td>
<td>88.9 ± 7.3</td>
<td>39 ± 9</td>
<td>-0.3 10</td>
</tr>
<tr>
<td>MONKS-1</td>
<td>91.1 ± 4.6</td>
<td>203 ± 42</td>
<td>-0.5 10</td>
</tr>
<tr>
<td>MONKS-2</td>
<td>77.8 ± 4.4</td>
<td>294 ± 9</td>
<td>-0.3 10</td>
</tr>
<tr>
<td>MONKS-3</td>
<td>88.3 ± 5.4</td>
<td>171 ± 46</td>
<td>-0.6 10</td>
</tr>
<tr>
<td>MUX-20</td>
<td>55.7 ± 6.3</td>
<td>388 ± 14</td>
<td>-0.1 10</td>
</tr>
<tr>
<td>NURSERY</td>
<td>77.2 ± 5.9</td>
<td>3271 ± 551</td>
<td>-0.9 10</td>
</tr>
<tr>
<td>SCALE</td>
<td>72.2 ± 4.2</td>
<td>394 ± 11</td>
<td>0.1 0</td>
</tr>
<tr>
<td>SPELCE</td>
<td>61.1 ± 3.7</td>
<td>1977 ± 101</td>
<td>-0.6 10</td>
</tr>
<tr>
<td>TIC-TAC</td>
<td>72.3 ± 4.8</td>
<td>468 ± 34</td>
<td>-0.4 10</td>
</tr>
<tr>
<td>VOTING</td>
<td>89.2 ± 5.8</td>
<td>52 ± 12</td>
<td>-0.3 10</td>
</tr>
<tr>
<td>WINE</td>
<td>78.6 ± 10.2</td>
<td>73 ± 12</td>
<td>-0.3 10</td>
</tr>
<tr>
<td>XOR-5</td>
<td>50.7 ± 10.6</td>
<td>136 ± 8</td>
<td>-0.1 10</td>
</tr>
<tr>
<td>ZOO</td>
<td>90.0 ± 7.4</td>
<td>24 ± 5</td>
<td>-0.2 10</td>
</tr>
</tbody>
</table>
Smaller trees have additional advantages:
- Comprehensibility
- Statistical significance at the leaves
- Low usage and storage costs

Greedy Split Decisions

Select a feature that separates best (locally)

Greedy Tree Induction

- Fast
- Local heuristics are good for simple tasks
- Fail when attribute inter-dependencies involved
- Problem intensified when costs are involved

Finding an Optimal Tree?

Consistent Trees

NP-complete!
Anytime: Between Greedy and Exponential

- Learning a tree greedily takes few seconds
- Learning an optimal tree is not practical

Main idea: Prefer feature with smallest consistent tree

Estimate the size of the minimal tree by sampling the population of trees under it

The size of the sample is determined by the contract

**LSID3: Contract Learning**

```
| | T | | T | | T |
|---|---|---|---|---|
| f1 | 1 | f1 | 2 | f2 | 3 |
| | 3 | | 5 | =5 |
| | 7 | | 5 | =5 |
| evaluation(f1) = min(5,3,7) = 3 | evaluation(f1) = min(5,5,7) = 5 |
```

**How to sample?**

- RCTG: many very large trees in the sample
- ID3: the same tree would be repeatedly induced
- Stochastic ID3: semi-random
- Probability proportional to gain
Contract vs. Interruptible

- In LSID3 the sample size is predetermined according to the contract time
- Large Samples $\Rightarrow$ better estimations
- Offline learning
- Sometimes allocation is not known in advance:
  - Learn until a classifier is needed
  - Learn until accuracy on a set-aside data is reached

IIDT: Interruptible Learning

- Build initial greedy tree
- Choose subtree with highest marginal utility
- Rebuild subtree with doubled resources
- Replace if Better
- Allows incremental learning

Anytime Learning: Results

- LSID3 and IIDT trees are smaller & more accurate

Statistical Significance

- Paired t-test wins:
- Wilcoxon test: the advantage of LSID3 and LSID3-p is significant
**Anytime Behavior**

- Good anytime behavior
- Difficult concepts, e.g., n-XOR, become learnable

![Graph](Image)

**Anytime Induction of Low-error, Low-cost Trees**

**Total Cost of Classification**

- The total cost during classification is the sum of:
  - If not given in the same scale: use a conversion

```
<table>
<thead>
<tr>
<th></th>
<th>true</th>
<th>predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0$</td>
<td>2$</td>
</tr>
<tr>
<td></td>
<td>7$</td>
<td>0$</td>
</tr>
<tr>
<td>Misclassification costs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing costs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>1$</td>
<td></td>
</tr>
<tr>
<td>gender</td>
<td>1$</td>
<td></td>
</tr>
<tr>
<td>BP</td>
<td>2$</td>
<td></td>
</tr>
<tr>
<td>HDL</td>
<td>4$</td>
<td></td>
</tr>
</tbody>
</table>
```

**Trading Learning Costs for Classification Costs**

![Graph](Image)
ACT

- Exploit extra time to induce lower-cost trees
- Build a tree top-down
- Sample trees under splits:
  - Evaluate by estimating the minimal cost of a subtree
  - Interruptible by iterative refinement as IIDT

\[ f_2 \]
\[ \epsilon \]
\[ \$55 \]
\[ \$47 \]
\[ \$81 \]
\[ \text{evaluation}(f_2) = \min(55, 47, 81) = 47 \]

ACT: Obtaining the Sample

- SEG2: A stochastic version of the EG2 algorithm
- EG2 builds a tree top-down and prefers at each node the attribute that maximizes:

- In SEG2 a split is selected semi-randomly: proportionally to ICF

ACT: Evaluating Trees

- The cost of a tree is the sum of 2 components:
  - Testing costs: approximated by the average testing cost on the training data
  - Misclassification cost: estimated using the expected error of the tree, multiplied by the error cost

\[ \text{cost-insensitive} \]
\[ \text{error} \]
Experiments

- ACT has been compared to:
  - C4.5, EG2, CSID3, IDX, DTMC and ICET

- Only 5 datasets in the UCI repository have costs
  - We designed a parametric method for automatic adaption of datasets to cost-sensitive environments
  - In total we used $(5 + 25^4)$ datasets

Fixed-time Results

Anytime Results

Anytime Induction of Anycost Classifiers
Different Cost Scenarios

Need for Special Solution?
- Feature subset?
  - Unknown bounds, irrelevant features
  - Tree-based classifiers?
    - Anycost by nature (store defaults)
    - C4.5 / LSID3: may require unaffordable tests
    - ICET / ACT: minimize total cost
    - TATA (Tree-classification AT Anycost)

TDIDT$
- Build a tree top-down
- Filter out tests whose costs exceed the budget
- Choose a split
- Call recursively, reduce current cost from budget

Remaining cost: $65
f3 $50
f2 $80
f1 $20
Remaining cost: $15

Pre-Contract TATA
- Instantiating TDIDT$ by how to choose a split
  - C4.5$: using info gain (greedy)
- Pre-Contract TATA
  - Use TDIDT$
  - Sample subtrees under candidate splits using SC4.5$
  - Evaluate sample-trees using expected error cost
**Pre-Contract Results**

![Graph showing classification cost and misclassification cost for different algorithms and parameters](image)

- **C4.5**, **EG2**, **EG2**$, **TATA(r=0)**, and **TATA(r=5)**

**Repertoires**

- **TDIDT**$ algorithms require $\rho$
- In contract and interruptible $\rho$ is not available to the learner
- Solution: repertoire of Pre-Contract-TATA trees
  - Several trees, each with a different $\rho$

**Contract Classification**

- **Learning**: $k$ trees, uniformly distribute $\rho$ values
  - $k = 3$
  - $\rho = 80$
  - $f_1$ $20$
  - $f_2$ $80$
  - $f_3$ $50$

  Actual $\rho$: 0 $20$ $65$ $130$

- **Classification**: pick the best-fit tree given $\rho$ (say $80$)

**Better $\rho$ Distribution?**

- **Problems with uniform distribution of $\rho$ values**:
  - Interruptible learning is impossible (if $k$ is unknown)
  - Some $\rho$ ranges may result in identical trees

<table>
<thead>
<tr>
<th>$\rho_{\text{min}}$</th>
<th>$\rho_{\text{min}} + \Delta$</th>
<th>$\rho_{\text{min}} + 2\Delta$</th>
<th>$\rho_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0$</td>
<td>$1$</td>
<td>$6$</td>
<td>$11$</td>
</tr>
<tr>
<td>$f_1$</td>
<td>$1$</td>
<td>$11$</td>
<td>$16$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>$1$</td>
<td>$11$</td>
<td>$16$</td>
</tr>
<tr>
<td>$f_3$</td>
<td>$1$</td>
<td>$11$</td>
<td>$16$</td>
</tr>
<tr>
<td>$f_4$</td>
<td>$13$</td>
<td>$16$</td>
<td>$16$</td>
</tr>
</tbody>
</table>

*Useless!*
**The Hill Climbing Approach**

Expected Error vs. Bouquet Expected Error

- \( \rho_{\text{min}} \)
- \( \rho_{\text{max}} \)

**More Trees Vs. Better Trees**

Misclassification cost vs. Sample size (n)

- \( T=1 \)
- \( T=3 \)
- \( T=5 \)

**Contract Results**

Misclassification cost vs. Maximal classification cost

- C4.5
- Uni-TATA\((r=0, k=16)\)
- Uni-TATA\((r=3, k=16)\)
- Hill-TATA\((r=3, k=16)\)

**Interruptible Classification**

- Neither the learner nor the classifier aware to \( \rho \)
- Basic approach: traverse until interrupted
Improved Interruptible

- Discount tests that appear in previous trees
- Proportionally to their probability to be administrated
- Resources during classification are wasted
- Optimal scheduling

Major Contributions

✓ A novel framework for anytime learning of cost-sensitive classifiers
✓ Adaptable to any cost model / scenario
✓ Significantly outperforms existing learners when allocated extra time
  ✓ test costs are dominant
  ✓ misclassification costs are dominant
  ✓ testing costs are bounded