Agent-based Models
building reliable, intelligible classifiers

William La Cava

Kourosh Danai
Lee Spector*

Sara Silva+
Leonard Vanneschi&
Mauro Castelli&

+University of Lisbon, PT
&Nova University, PT
*Hampshire College, USA
Outline

• My research - briefly
• Lisbon
• Multiclass Classification
• Bald Eagle Agent Modelling
  – Preliminary Results
My Research

• Automatic model form development and adaptation
  – Machine learning techniques for understanding complex systems

• Wind Energy Applications
  – Wind Turbine Dynamics
  – Vortex Induced Vibration
  – Bald Eagle Behavior
Wind Turbine Dynamics

NREL CART 3
Closed-Loop Data Collection
Source: nrel.gov/wind

Control Design

Model Development
# Wind Turbine Dynamics

<table>
<thead>
<tr>
<th></th>
<th>Training / Validation VAF (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ELGP</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>98.7 / 98.7</td>
</tr>
<tr>
<td>$\omega$</td>
<td>98.6 / 98.6</td>
</tr>
<tr>
<td>$M_{FA}$</td>
<td>74.2 / 74.4</td>
</tr>
<tr>
<td>$M_{SS}$</td>
<td>72.7 / 72.2</td>
</tr>
<tr>
<td>$P$</td>
<td>99.9 / 99.9</td>
</tr>
</tbody>
</table>

\begin{align*}
\omega &= n_1 (\omega_{k-1} + e^{(n_2 \omega_{k-1})} T_G) \\
\Omega &= n_1 (\Omega_{k-1} - \sin \left( \frac{n_2}{t} \right) \sin \left( \frac{V_{k-1} T_G_{k-1}}{\Omega_{k-1}} \right) \\
M_{FA} &= n_1 (M_{FA_{k-1}} + n_2 \sin (T_G) (\ddot{x}_{FA} - \ddot{x}_{FA_{k-1}}) / V) \\
M_{SS} &= n_1 (M_{SS_{k-1}} + n_2 \sin (n_3 \psi) (\ddot{x}_{FA} - \ddot{x}_{FA_{k-1}}))
\end{align*}

Vortex Induced Vibration

Bald Eagle Behavior

• Agent-based Modeling
• Discrete behaviors -> classes
• Multi-class classification
LISBON
University of Lisbon

• LabMAg
  – Laboratory of agent modeling
  – Institute for Complexity Sciences
• Nova University
Breakfast
M2GP and M3GP

A Multi-dimensional Genetic Programming Approach for Multi-class Classification Problems

Vijay Ingallali¹, Sara Silva¹,²,³, Mauro Castelli⁴, and Leonardo Vanneschi⁴

¹ INESC-ID, IST, University of Lisbon, 1000-029 Lisbon, Portugal
² LabMAg, FCUL, University of Lisbon, 1749-016 Lisbon, Portugal
³ CISUC, Universidade de Coimbra, 3030-290 Coimbra, Portugal
⁴ ISEGI, Universidade Nova de Lisboa, 1070-312 Lisboa, Portugal

Abstract. Classification problems are of profound interest for the machine learning community as well as to an array of application fields. However, multi-class classification problems can be very complex, in particular when the number of classes is high. Although very successful in so many application domains, Genetic Programming is regarded as a good method to perform multi-class classification. In this paper we present a novel algorithm for tree based GP, that incorporates a multi-dimensional representation of the solution space in higher dimensions. This proposal is based on the foundations of addressing multi-class classification problems in lower dimensional spaces and may lead to further research in this direction. We test the new algorithm in a large set of benchmark problems from several different sources to demonstrate competitiveness against the most successful state-of-the-art classification methods.

M3GP – Multiclass Classification with GP

Luis Muñoz¹, Sara Silva²,³,⁴, and Leonardo Trujillo¹

¹ Tree-Lab, Instituto Tecnológico de Tijuana, B.C., México
² BioISI – Biosystems & Integrative Sciences Institute, Faculty of Sciences, University of Lisbon, Portugal
³ NOVA IMS, Universidade Nova de Lisboa, 1070-312 Lisboa, Portugal
⁴ CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
sara@fc.ul.pt, {lmunoz, leonardo.trujillo}@tectijuana.edu.mx

Abstract. Data classification is one of the most ubiquitous machine learning tasks in science and engineering. However, Genetic Programming is still not a popular classification methodology, partially due to its poor performance in multiclass problems. The recently proposed M2GP - Multidimensional Multi-class Genetic Programming algorithm achieved promising results in this area, by evolving mappings of the p-dimensional data into a d-dimensional space, and applying a minimum Mahalanobis distance classifier. Despite good performance, M2GP employs a greedy strategy to set the number of dimensions d for the transformed data, and fixes it at the start of the search, an approach that is prone to locally optimal solutions. This work presents the M3GP algorithm, that stands for M2GP with multidimensional populations. M3GP extends M2GP by allowing

- progress
Why use GP?

- Lots of classifier systems already exist (SVMs, MLPs, Decision trees, Random Forests)
- GP isn’t traditionally good at multi-class classification problems
- Less assumptions
- Intelligibility
- Generalizability
- Feature selection
Lunch
Insight

• Use GP to generate features
  – original attributes -> distinct distributions

• Mahalanobis distance
  \[ D_k = \sqrt{(X - M_k)C_k^{-1}(X - M_k)^T} \]
  – each transformed attribute vector to each class’ attribute distribution in new space

• Assign class with lowest \( D_k \)

• Classifier: Equations, \( M, C \)
Challenges

• The results were good, but not great
  – Tied or worse than Random Forests & other methods
• *Not making use of more recent advances to other parts of GP*
• *Complex tree representation*
Lots of methods

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tournament selection</td>
<td>tourn</td>
<td>size 2 standard tournaments.</td>
</tr>
<tr>
<td>pareto survival</td>
<td>ps</td>
<td>multi-objective age-fitness pareto survival via SPEA2</td>
</tr>
<tr>
<td>lexicase selection</td>
<td>lex</td>
<td>parents are selected using lexicase selection, where the cases are each data sample</td>
</tr>
<tr>
<td>class-based pareto survival</td>
<td>ps5</td>
<td>multiple fitnesses are assigned, one for each class label. age is also used and pareto survival is conducted via SPEA2</td>
</tr>
<tr>
<td>lexicase selection with classes</td>
<td>lexc</td>
<td>parents are selected using lexicase selection, where the cases are the aggregate error on each class label</td>
</tr>
</tbody>
</table>

I also tried three different settings for the genetic operators:

<table>
<thead>
<tr>
<th>Genetic Operators Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uniform</td>
<td>uniform alternation &amp; point mutation</td>
</tr>
<tr>
<td>st</td>
<td>subtree crossover and mutation, specialized according to m3gp</td>
</tr>
<tr>
<td>stp</td>
<td>subtree crossover and mutation plus pruning of the best individual each generation</td>
</tr>
</tbody>
</table>
Test Problems

- UCI Repository data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>heart</th>
<th>mcd10</th>
<th>mcd3</th>
<th>movl</th>
<th>seg</th>
<th>vowel</th>
<th>wav</th>
<th>yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes</td>
<td>2</td>
<td>10</td>
<td>3</td>
<td>15</td>
<td>7</td>
<td>11</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Attributes</td>
<td>13</td>
<td>6</td>
<td>6</td>
<td>90</td>
<td>19</td>
<td>13</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>Cases</td>
<td>270</td>
<td>6798</td>
<td>322</td>
<td>360</td>
<td>2310</td>
<td>990</td>
<td>5000</td>
<td>1484</td>
</tr>
</tbody>
</table>
Lots of Results

Validation Fitness

When we look at validation fitness, M3GP is often not the best.
Overall Results

Median Training Ranking

our system  Typical ML  old system
Overall Results

The chart compares the performance of different systems, including our system, a typical ML system, and an old system. The x-axis represents various datasets and models, while the y-axis shows the median test ranking. The bars indicate the performance, with colors representing different systems and datasets.
Bald Eagle

- 11,537 attributes
- 4 behaviors
  - Flight
  - Perched
  - Cruise
  - Nest
Results

Median classifier accuracy: 98.87%
Best classifier accuracy : 99.82%
Features

**speed:** separates [flight, cruise] from [nest, perched]

**agl:** separates [flight] from [cruise]

**nest dist:** separates [nest] from [perched]
Principal Component Analysis
Number of features
Future Work

• Try simpler classification schemes for Bald Eagle data

• New data
  – More specific behaviors
  – Classify new data automatically

• Interpretation of results

• Test scalability of GP method
Thank you!