Genetic Programming:
Data Mining Case Study

Andrew Kusiak
Intelligent Systems Laboratory, 3221 SC
The University of Iowa
Iowa City, IA 52242 - 1527
Tel. 319-335 5934
andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak

Data Mining: Introduction
Recognition of words from the set
\{0001, 0011, 0111, 1111\} by using:

- Decision table
- Decision tree
- Decision rules

Decision Table
Words: 0001, 0011, 0111, 1111

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>One</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Two</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Three</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Four</td>
</tr>
</tbody>
</table>

Decision Tree
Words: 0001, 0011, 0111, 1111

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>One</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Two</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Three</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Four</td>
</tr>
</tbody>
</table>

Decision Tree

- Rule 1. (F3 = 0) THEN (D = One);
  [1, 100.00%, 100.00%] [1]
- Rule 2. (F2 = 0) AND (F3 = 1) THEN (D = Two);
  [1, 100.00%, 100.00%] [2]
- Rule 3. (F1 = 0) AND (F2 = 1) THEN (D = Three);
  [1, 100.00%, 100.00%] [3]
- Rule 4. (F1 = 1) THEN (D = Four);
  [1, 100.00%, 100.00%] [4]
Rule Format

Rule 1. \((F3 = 0)\) THEN \((D = One)\);
\([1, 100.00\%, 100.00\%]\) [1]

(Condition) THEN (Outcome);

- Rule support
- Relative rule strength
- Discrimination level
- Objects represented by the rule

Decision Rules

Rule 1. \((F3 = 0)\) THEN \((D = One)\);
\([1, 100.00\%, 100.00\%]\) [1]
Rule 2. \((F2 = 0)\) AND \((F3 = 1)\) THEN \((D = Two)\);
\([1, 100.00\%, 100.00\%]\) [2]
Rule 3. \((F1 = 0)\) AND \((F2 = 1)\) THEN \((D = Three)\);
\([1, 100.00\%, 100.00\%]\) [3]
Rule 4. \((F1 = 1)\) THEN \((D = Four)\);
\([1, 100.00\%, 100.00\%]\) [4]

F1 F2 F3 F4 D
0 0 0 1 One
0 0 1 1 Two
0 1 1 1 Three
1 1 1 1 Four

Rule Tree

Decision Tree vs Rule Tree

Data Mining Example

No. F1 F2 F3 F4 D
1 0 1 0 2 Zero
2 1 1 0 2 Two
3 0 0 0 1 Zero
4 0 1 1 0 One
5 0 0 1 3 Zero

Classification Quality

Classification quality = Measure of association between a feature and the decision; a number in the interval \([0, 1]\)
Classification Quality

<table>
<thead>
<tr>
<th>No.</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>D</th>
<th>Classification quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>Zero</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>Two</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Zero</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>One</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>Zero</td>
</tr>
</tbody>
</table>

CQ (F1) = 1/5 = .2

Rule Set

1. (F2 = 0) THEN (D = Zero); [2, 66.67%, 100.00%] [3, 5]
2. (F1 = 0) AND (F3 = 0) THEN (D = Zero); [2, 66.67%, 100.00%] [1, 3]
3. (F4 = 0) THEN (D = One); [1, 100.00%, 100.00%] [4]
4. (F1 = 1) THEN (D = Two); [1, 100.00%, 100.00%] [2]

Absolute Classification Accuracy

1-out-of-n cross-validation scheme

Feature set {F1, F2, F3, F4} n = 5

<table>
<thead>
<tr>
<th>D</th>
<th>Zero</th>
<th>One</th>
<th>Two</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>One</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Two</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Classification Accuracy

1-out-of-n cross-validation scheme

Feature set {F1, F2, F3, F4}

<table>
<thead>
<tr>
<th></th>
<th>Correct</th>
<th>Incorrect</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero</td>
<td>33.33%</td>
<td>33.33%</td>
<td>33.33%</td>
</tr>
<tr>
<td>One</td>
<td>0.00%</td>
<td>100.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Two</td>
<td>0.00%</td>
<td>100.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Av</td>
<td>20.00%</td>
<td>60.00%</td>
<td>20.00%</td>
</tr>
</tbody>
</table>

Compound Feature

Definition

Compound feature = a sequence of features

Compound Feature F2_F3

<table>
<thead>
<tr>
<th>No.</th>
<th>F1</th>
<th>F2_F3</th>
<th>F4</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>_0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>_0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0_0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1_1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0_1</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
**Compounds Feature F2_F3**

### Classification Quality

<table>
<thead>
<tr>
<th>Feature set</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F1_F2</th>
<th>F1_F3</th>
<th>F1_F4</th>
<th>F2_F3</th>
<th>F2_F4</th>
<th>F3_F4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification quality</td>
<td>.2</td>
<td>.8</td>
<td>.6</td>
<td>.4</td>
<td>0</td>
<td>.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Rules

- **Rule 5.** (A2_3 = 0_0) THEN (D = Zero);
  - [1, 33.33%, 100.00%][3]
- **Rule 6.** (A2_3 = 0_1) THEN (D = Zero);
  - [1, 33.33%, 100.00%][5]
- **Rule 7.** (A1 = 0) AND (A4 = 2) THEN (D = Zero);
  - [1, 33.33%, 100.00%][1]
- **Rule 8.** (A2_3 = 1_1) THEN (D = One);
  - [1, 100.00%, 100.00%][4]
- **Rule 9.** (A1 = 1) THEN (D = Two);
  - [1, 100.00%, 100.00%][2]

---

**Genetic Programming**

**Problem Formulation**

Given a training data set, find a set of feature maximizing the classification accuracy.
Genetic Programming

Create random initial population

Evaluate population

Insert into population

Vary

Evaluate population

Select

Solution Representation

F1 F2 F3 F4

How to define
• Mutation
• Crossover?

Case 1

Case 2

Solution Representation

Genotype

0011 1 0 0 0 0 1 0 . . .

Phenotype

F1 F2 F3 F4 F1_F2 F1_F3 F1_F4 F2_F3 F2_F4 F3_F4 F1_F2_F3 ...

1 - Feature present
0 - Feature absent

Case 2

Mutation

Before

0 0 1 1 1 0 0 0 0 1 0 . . .

After

0 0 1 1 1 0 0 0 0 1 0 . . .

Case 2

Crossover

1-point crossover

Chromosome 1

0 0 1 1 1 0 0 0 0 1 0 . . .

Chromosome 2

1 1 1 1 1 0 0 1 1 0 . . .

Reproduction

‘Original’

0 0 1 1 1 0 0 0 0 1 0 . . .

Copy the chromosome
Solution Representation

\[ F_1 \ F_2 \ F_1F_2 \ F_1F_3 \ F_1F_4 \ F_2F_3 \ F_1F_2F_3 \ldots \]

Mutation

Before

\[ F_1 \ F_2 \ F_1F_2 \ F_1F_3 \ F_1F_4 \ F_2F_3 \ F_1F_2F_3 \ldots \]

After

\[ F_1 \ F_2 \ F_1F_2 \ F_2F_3 \ F_1F_4 \ F_1F_3F_4 \ F_1F_2F_3 \ldots \]

Crossover

\[ F_1 \ F_2 \ F_3 \ F_4 \]

Preferred Representation

\[ \text{Case 22? If NO Case 4} \]

Fitness Measure

- Classification quality
  - not obvious candidate
- Classification accuracy
  - good candidate

Fitness Measure

Classification accuracy =

\[ f \text{ (Learning algorithm, Data set, Feature Set)} \]
Fitness Evaluation

- Genotype (corresponds to a set of features)
- Learning Algorithm
- Cross-validation procedure
- Fitness (= Classification accuracy)

Fitness Function Challenge

Define a fitness function that could be computed based on the information contained in a chromosome

Genotype - Phenotype Transformation

**IF**
Genotype = String of binary numbers

**AND**
Phenotype = Set of features

**THEN**
Transformation =
Substitutions of binary numbers with features

Genotype-Phenotype Transformation

Genotype

Phenotype

Genetic Programming Algorithm

Terminal Set:
Basic and compound features, each of size not greater than 3

Function Set:
‘ADD’, ‘SUB’

GP Algorithm

- Population size, 10
- Initialization method: Grow
- Crossover probability, .9
- Mutation probability, .05
- Selection method: tournament, size 4
- Maximum length of a chromosome, \( l = 12 \)
Genetic Programming Algorithm

- Number of generations
- Classification accuracy gain per, e.g., 10 generations <= 1%

Architecture Alteration

‘Revision’ of the Final Chromosome (Solution)

E.g., Features that do not appear in the rules derived by a learning algorithm considered in the study are to be removed.

Recommended Reading

- Example on pages 135 - 141 of the textbook
- Chapter 12: GP Applications

References

