Random Forest Ensemble Visualization

CPSC 547 Project
Ken Lau
Prediction

Data ➤ Machine learning algorithm ➤ Prediction
## Weather Data Example

### Feature Variables

<table>
<thead>
<tr>
<th>Obs</th>
<th>Avg rain today</th>
<th>Humidity today</th>
<th>Sunny percentage today</th>
<th>Weather Tomorrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1mm</td>
<td>81 %</td>
<td>10%</td>
<td>Rainly</td>
</tr>
<tr>
<td>2</td>
<td>2mm</td>
<td>83 %</td>
<td>40%</td>
<td>Cloudy</td>
</tr>
<tr>
<td>3</td>
<td>0 mm</td>
<td>80 %</td>
<td>80%</td>
<td>Sunny</td>
</tr>
<tr>
<td>4</td>
<td>1 mm</td>
<td>81 %</td>
<td>20 %</td>
<td>Cloudy</td>
</tr>
</tbody>
</table>
Classification Tree

Input Feature Variables

Avg rain today > 2mm

Humidity today > 81%

sunny

Sunny percentage > 10%

Tomorrow cloudy

Tomorrow sunny

Tomorrow rainy
Random Forest

• Collection of classification trees
  – Usually 500-1000

• Popular

• Black box
Random Forest

Data → Random Sample → Sample 1
<table>
<thead>
<tr>
<th></th>
<th>Sample 2</th>
<th>Tree 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sample m</td>
<td>Tree m</td>
</tr>
</tbody>
</table>

Majority Vote → Prediction
My Data

- $r[n]$
  - No: $m_1 > 55$
    - No: $m_3 > 0.23$
      - 6
    - Yes: $m_5 > 0.74$
      - 3
  - Yes: $m_2 > 0.51$
    - 1
    - 2
    - 5
    - 4
Problem:
How to visualize the collection of Classification trees
Aggregate: Features Variables

Tree 1

- Depth: 1
- Feature: m1
- Appearance: 2

Tree 2

- Depth: 1
- Feature: m1
- Left Split: 3
- Right Split: 4

Depth | Parent | Feature | Left Split | Right Split
--- | --- | --- | --- | ---
2 | m1 | m2 | 1 | 1
2 | m1 | m1 | 1 | 0

Encode Colour Saturation

Data Derive
Aggregate: Class Prediction Variables

Tree 1

Tree 2

<table>
<thead>
<tr>
<th>Depth</th>
<th>Feature</th>
<th>Cl 1</th>
<th>Cl 2</th>
<th>Cl 3</th>
<th>Cl 4</th>
<th>Cl 5</th>
<th>Cl 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>m1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Depth</th>
<th>Parent</th>
<th>Feature</th>
<th>Cl 1</th>
<th>Cl 2</th>
<th>Cl 3</th>
<th>Cl 4</th>
<th>Cl 5</th>
<th>Cl 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>m1</td>
<td>m2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Depth</th>
<th>Parent</th>
<th>Feature</th>
<th>Cl 1</th>
<th>Cl 2</th>
<th>Cl 3</th>
<th>Cl 4</th>
<th>Cl 5</th>
<th>Cl 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>m1</td>
<td>m1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Visualization

- So What?
  - Feature importance and interaction
  - Tree pruning when non-uniform class count distribution occurs
  - Class count predictions given nodes traversed so far
Software

• Python
  – Model fitting
  – Information retrieval
  – Aggregation

• D3
  – Encoding
  – Based on Indented Tree (Mike Bostock, 2011)
    http://bl.ocks.org/mbostock/1093025
Demo

Visualization link:
http://kenlau177.github.io/Indented-Agg-Tree/
Scale

- Manageable up to trees of depth 8 with 5 feature variables.
  - Out of memory issue
  - There is a step that generates all possible permutations of features variables
  - Instead keep only variables that appear at least once in the collection of trees

- Handles more than 1000 trees fast with depth less than 7

<table>
<thead>
<tr>
<th>Number of Trees</th>
<th>Depth</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>7</td>
<td>59 sec</td>
</tr>
<tr>
<td>800</td>
<td>3</td>
<td>10 sec</td>
</tr>
<tr>
<td>1500</td>
<td>3</td>
<td>15 sec</td>
</tr>
<tr>
<td>1500</td>
<td>6</td>
<td>22 sec</td>
</tr>
</tbody>
</table>
Quantify the Tree Ensembles

• Measure diversity among trees based on class predictions
• Unrelated members are the reason for high accuracy
• Hamann Similarity Measure
  – Multivariate version
### Predicted **Same** Class

<table>
<thead>
<tr>
<th>Tree 1</th>
<th>Tree 2</th>
<th>Correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>a1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Incorrect</td>
<td>0</td>
<td>d1</td>
<td></td>
</tr>
</tbody>
</table>

### Predicted **Different** Class

<table>
<thead>
<tr>
<th>Tree 1</th>
<th>Tree 2</th>
<th>Correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>0</td>
<td>b2</td>
<td></td>
</tr>
<tr>
<td>Incorrect</td>
<td>c2</td>
<td>d2</td>
<td></td>
</tr>
</tbody>
</table>

**Equation:**

\[
H = \frac{(a_1 + d_1) - (b_2 + c_2 + d_2)}{a_1 + d_1 + b_2 + c_2 + d_2}
\]
Tree 1

Tree 2

... 

Tree 500

Derived Data

Histogram of Hamann
Similarity of Tree Ensembles

count

Hamann Similarity
Filter Trees based on Hamann Similarity
Filter Trees based on Hamann Similarity

Histogram of Hamann Similarity of Tree Ensembles

Variable Importance

Multi-dimensional scaling

Hamann Range

.58 - .76
Thank you

R Shiny App:
https://kenlau177.shinyapps.io/randomForestApp/