Ensembles
Making Trees Unstoppable

Poul Petersen
CIO, BigML, Inc
What is an Ensemble?

- Rather than build a single model...
- Combine the output of several typically “weaker” models into a powerful ensemble...
- Q1: Why is this necessary?
- Q2: How do we build “weaker” models?
- Q3: How do we “combine” models?
No Model is Perfect

• A given ML algorithm may simply *not be able* to exactly model the “real solution” of a particular dataset.

• Try to fit a line to a curve
Simple Example - Fit a Line
Simple Example - Fit a Line
No Model is Perfect

- A given ML algorithm may simply **not be able** to exactly model the “real solution” of a particular dataset.
  - Try to fit a line to a curve
- Even if the model is very capable, the “real solution” may be elusive
  - DT/NN can model any decision boundary with enough training data, but the solution is NP-hard
  - Practical algorithms involve random processes and may arrive at different, yet equally good, “solutions” depending on the starting conditions, local optima, etc.
- If that wasn’t bad enough…
No Data is Perfect

• Not enough data!
  • Always working with finite training data
  • Therefore, every “model” is an approximation of the “real solution” and there may be several good approximations.

• Anomalies / Outliers
  • The model is trying to generalize from discrete training data.
  • Outliers can “skew” the model, by overfitting

• Mistakes in your data
  • Does the model have to do everything for you?
  • But really, there is always mistakes in your data
Ensemble Techniques

- **Key Idea:**
  - By combining several good “models”, the combination may be closer to the best possible “model”
  - We want to ensure diversity. It’s not useful to use an ensemble of 100 models that are all the same

- Training Data Tricks
  - Build several models, each with only some of the data
  - Introduce randomness directly into the algorithm
  - Add training weights to “focus” the additional models on the mistakes made

- Prediction Tricks
  - Model the mistakes
  - Model the output of several different algorithms
Simple Example - Fit a Line

Partition the data… then model each partition…

For predictions, use the model for the same partition
Decision Forest

DATASET

SAMPLE 1

SAMPLE 2

SAMPLE 3

SAMPLE 4

MODEL 1

MODEL 2

MODEL 3

MODEL 4

PREDICTION 1

PREDICTION 2

PREDICTION 3

PREDICTION 4

COMBINER

PREDICTION
Things are getting harder to configure...

- Individual tree parameters are still available
  - Balanced objective, Missing splits, Node Depth, etc.
- **Number of models**: How many trees to build
- Sampling options:
  - **Deterministic / Random**
  - **Replacement**:
    - Allows sampling the same instance more than once
    - Effectively the same as $\approx 63.21\%$
    - “Full size” samples with zero covariance (good thing)
- At prediction time
  - Combiner…
# Outlier Example

What is a round, red 6cm fruit?

<table>
<thead>
<tr>
<th>Diameter</th>
<th>Color</th>
<th>Shape</th>
<th>Fruit</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>red</td>
<td>round</td>
<td>plum</td>
</tr>
<tr>
<td>5</td>
<td>red</td>
<td>round</td>
<td>apple</td>
</tr>
<tr>
<td>5</td>
<td>red</td>
<td>round</td>
<td>apple</td>
</tr>
<tr>
<td>6</td>
<td>red</td>
<td>round</td>
<td>plum</td>
</tr>
<tr>
<td>7</td>
<td>red</td>
<td>round</td>
<td>apple</td>
</tr>
</tbody>
</table>

All Data: “plum”

Sample 1: “plum”

Sample 2: “apple”

Sample 3: “apple”
Random Decision Forest

- **DATASET**
  - **SAMPLE 1**
    - **MODEL 1**
      - **PREDICTION 1**
  - **SAMPLE 2**
    - **MODEL 2**
      - **PREDICTION 2**
  - **SAMPLE 3**
    - **MODEL 3**
      - **PREDICTION 3**
  - **SAMPLE 4**
    - **MODEL 4**
      - **PREDICTION 4**

- **COMBINER**
  - **PREDICTION**
RDF Config

Things are getting harder to configure...

- Individual tree parameters are still there…
  - Balanced objective, Missing splits, Node Depth, etc.
- Decision Forest parameters still available
  - Number of model, Sampling, etc
- Random candidates:
  - The number of features to consider at each split
"Hey Model 1, what do you predict is the sale price of this home?"

"Hey Model 2, how much error do you predict Model 1 just made?"

**Why stop at one iteration?**
Boosting

Iteration 1
- DATASET 1
- MODEL 1
- PREDICTION 1

Iteration 2
- DATASET 2
- MODEL 2
- PREDICTION 2

Iteration 3
- DATASET 3
- MODEL 3
- PREDICTION 3

Iteration 4
- DATASET 4
- MODEL 4
- PREDICTION 4

SUM

Prediction
Wait a Second…

... what about classification?

... we could try
Wait a Second…

…but then what about multiple classes?

<table>
<thead>
<tr>
<th>pregnancies</th>
<th>plasma glucose</th>
<th>blood pressure</th>
<th>triceps skin thickness</th>
<th>insulin</th>
<th>bmi</th>
<th>diabetes pedigree</th>
<th>age</th>
<th>favorite color</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>148</td>
<td>72</td>
<td>35</td>
<td>0</td>
<td>33.6</td>
<td>0.627</td>
<td>50</td>
<td>RED</td>
</tr>
<tr>
<td>1</td>
<td>85</td>
<td>66</td>
<td>29</td>
<td>0</td>
<td>26.6</td>
<td>0.351</td>
<td>31</td>
<td>GREEN</td>
</tr>
<tr>
<td>8</td>
<td>183</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td>23.3</td>
<td>0.672</td>
<td>32</td>
<td>BLUE</td>
</tr>
<tr>
<td>1</td>
<td>89</td>
<td>66</td>
<td>23</td>
<td>94</td>
<td>28.1</td>
<td>0.167</td>
<td>21</td>
<td>RED</td>
</tr>
</tbody>
</table>

Model 1

Predicted favorite color
- BLUE
- GREEN
- RED
- ERROR

...but then what about multiple classes?
Boosting Classification

**Iteration 1**

<table>
<thead>
<tr>
<th>pregnancies</th>
<th>plasma glucose</th>
<th>blood pressure</th>
<th>triceps skin thickness</th>
<th>insulin</th>
<th>bmi</th>
<th>diabetes pedigree</th>
<th>age</th>
<th>favorite color</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>148</td>
<td>72</td>
<td>35</td>
<td>0</td>
<td>33.6</td>
<td>0.627</td>
<td>50</td>
<td>RED</td>
</tr>
<tr>
<td>1</td>
<td>85</td>
<td>66</td>
<td>29</td>
<td>0</td>
<td>26.6</td>
<td>0.351</td>
<td>31</td>
<td>GREEN</td>
</tr>
<tr>
<td>8</td>
<td>183</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td>23.3</td>
<td>0.672</td>
<td>32</td>
<td>BLUE</td>
</tr>
<tr>
<td>1</td>
<td>89</td>
<td>66</td>
<td>23</td>
<td>94</td>
<td>28.1</td>
<td>0.167</td>
<td>21</td>
<td>RED</td>
</tr>
</tbody>
</table>

**Class RED Probability**
- 0.9
- 0.7
- 0.46
- 0.12

**Class RED ERROR**
- 0.1
- -0.7
- 0.54
- -0.12

**Class BLUE Probability**
- 0.1
- 0.3
- 0.54
- 0.88

**Class BLUE ERROR**
- -0.1
- 0.7
- -0.54
- 0.12

...and repeat for each class at each iteration

**Iteration 2**

<table>
<thead>
<tr>
<th>pregnancies</th>
<th>plasma glucose</th>
<th>blood pressure</th>
<th>triceps skin thickness</th>
<th>insulin</th>
<th>bmi</th>
<th>diabetes pedigree</th>
<th>age</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>148</td>
<td>72</td>
<td>35</td>
<td>0</td>
<td>33.6</td>
<td>0.627</td>
<td>50</td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>85</td>
<td>66</td>
<td>29</td>
<td>0</td>
<td>26.6</td>
<td>0.351</td>
<td>31</td>
<td>-0.7</td>
</tr>
<tr>
<td>8</td>
<td>183</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td>23.3</td>
<td>0.672</td>
<td>32</td>
<td>0.54</td>
</tr>
<tr>
<td>1</td>
<td>89</td>
<td>66</td>
<td>23</td>
<td>94</td>
<td>28.1</td>
<td>0.167</td>
<td>21</td>
<td>-0.12</td>
</tr>
</tbody>
</table>

**PREDICTED ERROR**
- 0.05
- -0.54
- 0.32
- -0.22

...and repeat for each class at each iteration
Boosting Classification

**Iteration 1**
- **DATASET**
- **MODELS 1** per class
- **PREDICTIONS 1** per class

**Iteration 2**
- **DATASETS 2** per class
- **MODELS 2** per class
- **PREDICTIONS 2** per class

**Iteration 3**
- **DATASETS 3** per class
- **MODELS 3** per class
- **PREDICTIONS 3** per class

**Iteration 4**
- **DATASETS 4** per class
- **MODELS 4** per class
- **PREDICTIONS 4** per class

**Comb**
- **PROBABILITY** per class

**Ensembles**
Boosting Config

Things are getting harder to configure...

- **Number of iterations** - similar to number of models for DF/RDF
- Iterations can be limited with **Early Stopping**:
  - **Early out of bag**: tests with the out-of-bag samples
  - **Early holdout**: tests with a portion of the dataset
  - **None**: performs all iterations. Note: In general, it is better to use a high number of iterations and let the early stopping work.
- **Learning Rate**: Controls how aggressively boosting will fit the data:
  - Larger values ~ maybe quicker fit, but risk of overfitting
- You can combine sampling with Boosting!
  - Samples with Replacement
  - Add Randomize
- Individual tree parameters are still available
  - Balanced objective, Missing splits, Node Depth, etc.
Ensembles Demo
Which Ensemble Method

- For "large" / "complex" datasets
  - Use DF/RDF with deeper node threshold
  - Even better, use Boosting with more iterations
- For "noisy" data
  - Boosting may overfit
  - RDF preferred
- For "wide" data
  - Randomize features (RDF) will be quicker
- For "easy" data
  - A single model may be fine
  - Bonus: also has the best interpretability!
- For classification with "large" number of classes
  - Boosting will be slower
- For "general" data
  - DF/RDF likely better than a single model or Boosting.
  - Boosting will be slower since the models are processed serially

**Real Answer: Use the one that works best!**
- Ok, but seriously. Did you evaluate?
Your Turn!

- Pick ONE of the three Ensemble methods
- Create an ensemble for the Diabetes 80% Training dataset
- Evaluate the Ensemble with the 20% Test set.
- Compare the Evaluation with the Model you made earlier
- Which performs better?