Leaf Classification

IDENTIFYING PLANT SPECIES BY THE CHARACTERISTICS OF THEIR LEAVES

AUTHOR: ZACH STECHER

Project Details

- The point of this project was to produce a submission to the Leaf Classification Kaggle competition.
- Participants were supplied with a training and testing data set, as well as pictures of each leaf.
- The objective was to find a combination of classifier and data attributes that could accurately predict a plant's species based on the shape of its leaf.
- Score determined by submission's Log Loss. Better scores approach zero.

Why this project?

- A decent introduction to image recognition applications of artificial intelligence.
- Possible applications
  - Crop management
  - Medical practice in areas without ready access to modern medicine
  - Easier environmental tracking and impact study

Methods

- Ran a classifier comparison program available on the Kaggle competition forum.
  - This provided a solid starting idea of which classifiers performed and which weren't worth looking at.
- Also attempted to run TensorFlow and Theano.
- Had to fall back on RandomForest as a benchmark when TensorFlow and Theano did not pan out due to technical issues.
- Picked two classifiers from the comparison to go against RandomForest to find the most promising of the three.

Classifiers

- Tried KNeighborClassifier and Linear Discriminant Analysis as they provided the best initial scores.
- RandomForest generated the best score of the three “out of the box”.
- Attempted to tune Kneighbor and LDA to beat RandomForest’s initial score.
- After both classifiers failed to beat RandomForest, I decided to try and get the best score with RandomForest.

RandomForest Details

- An ensemble learning algorithm
- Utilizes multiple “decision tree” classifiers
- Randomly assigns each of these decision tree classifiers a portion of the data, but never all of it.
- Also randomizes which attributes are considered at which trees, to avoid giving too much weight to possibly irrelevant attributes
- Averages the responses from every decision tree classifier to come up with an answer

Experiments

- Tried two different RandomForest implementations and compared their results at the same number of trees
- One implementation utilized Stratified Shuffle Split cross validation, while the other did not utilize cross validation at all

Results

- Best score achieved by KNeighbors: 1.63982
- Best score achieved by LinearDiscriminationAnalysis: 1.24583
- Best score achieved by RandomForest with Cross Validation: 0.77641
  - Multiple runs at differing numbers of trees all achieved results between 0.77641 and ~0.78000
- Best score achieved by RandomForest without CV: 0.67886
  - Multiple runs at differing numbers of trees all achieved results between 0.67886 and ~0.68500

Observations

- RandomForest seems to be the best jumping off point for this problem due to it's resistance toward providing undue weight to irrelevant features. This is especially important in image recognition where features may vary in importance from image to image.
- RandomForest also seems to be highly resistant to overfitting, as once overfitting DID begin at high tree levels, the accuracy on the test data decreased very slowly.
- Feature selection is much more important than I initially thought and is likely the biggest reason I could not get my best score lower.

In The Future

- Would still like to try a proper neural network on this problem.
  - Especially fix technical issues with TensorFlow and Theano and try them
- Other submissions dabbled with things like Keras Neural Networks and feature extraction from the images themselves rather than the provided .csv data set.
- With enough time, would be interesting to see if proper feature choice and attribute tuning on unexpected classifier would yield better results

Questions?

- Examples:
  - Why are the Jets terrible this year?
  - Did you watch the Westworld finale?
  - How many support vectors does it take to change a light bulb?