18. Decision Trees

Bruce E. Shapiro

Getting Started in Machine Learning

Last revised: March 24, 2019

Splitting Criterion

- **RSS Error** (as in regression trees)
- **Information Gain** $\Delta I = -\sum_i \frac{|S_i|}{|S|} S_i$ where $S = -\sum_i p_i \log_2 p_i$ is the entropy, p_i is the proportion assigned to each class, and we define $p \log p = 0$ if $p = 0$.
- **Gini Impurity** $G = 1 - \sum_{i=1}^{K} p_i^2$
- **Chi squared** $\chi^2 = \sum_i \frac{(y_i - \hat{y}_i)^2}{\hat{y}_i}$ A higher chi-squared value gives more information.

Stopping Condition

- (a) Tree Depth
  - if not specified or too large, may continue splitting until only one data point in each leaf
  - if too small may have many data points in each leaf
- (b) Number of data points in leaves
  - If too large, trees may not get very deep, won't discriminate well
- Combination of (a) and (b)

Python Example - Car Cylinder Discrimination (1/4)

Import data set from UCI database

```
import pandas as pd
data=pd.read_fwf("https://archive.ics.uci.edu/ml/
  machine-learning-databases/auto-mpg/auto-mpg.data",
  header=None,na_values="?")
data.columns=("mpg","cyl","displ","hp","weight",
  "accel","model","origin","carname")
data = data.dropna(axis=0)
```

Python Example - Car Cylinder Discrimination (1/4)

Import data set from UCI database

```
import pandas as pd
data=pd.read_fwf("https://archive.ics.uci.edu/ml/
  machine-learning-databases/auto-mpg/auto-mpg.data",
  header=None,na_values="?")
data.columns=("mpg","cyl","displ","hp","weight",
  "accel","model","origin","carname")
data = data.dropna(axis=0)
```

Extract 5 features, 3 classes (4, 6, 8 cylinders)

```
import numpy as np
cars=np.array(data[["cyl","mpg","displ","hp","weight",
   "accel"]])
cars=np.array([line for line in cars
   if line[0] in [4,6,8]])
Y=cars[:,0]/2-2
X=cars[:,1:]
```

Python Example - Car Cylinder Discrimination (2/4)

```
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
nsplits=100; depth=3; errs=[]
for j in range(nsplits):
    XTRAIN, XTEST, YTRAIN, YTEST=train_test_split(X,Y)
    DT=tree.DecisionTreeClassifier(max_depth=depth)
    DT.fit(XTRAIN, YTRAIN)
    YP=DT.predict(XTEST)
    errs.append(1-accuracy_score(YTEST,YP))
print("Decision Tree Depth = %d mean error =
    %7.6f SD = %7.6f"\
    %(depth,np.mean(errs),np.std(errs)))
```

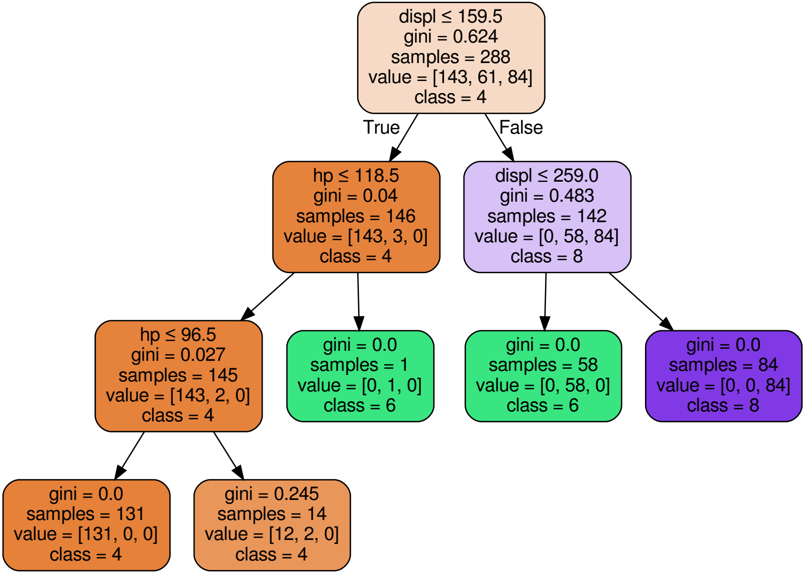
```
Decision Tree Depth = 3 mean error = 0.020412 SD = 0.014651
```

Python Example - Car Cylinder Discrimination (3/4)

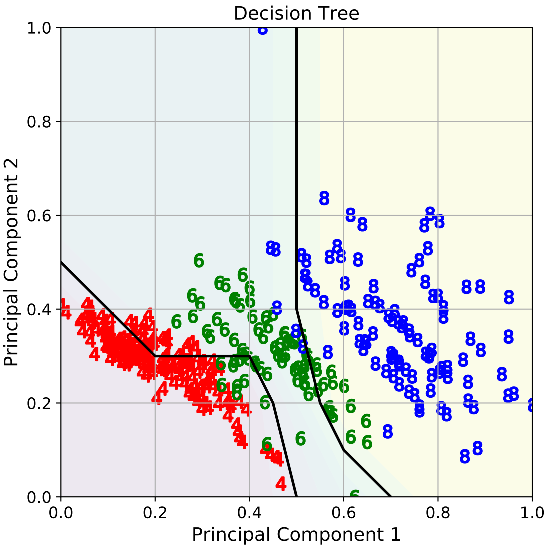
```
import graphviz, pydotplus

dot_data = tree.export_graphviz(DT, out_file=None,
   rotate=False,
   feature_names=["mpg","displ","hp","weight","accel"],
   class_names=list(map(str,[4,6,8])),
   filled=True, rounded=True, special_characters=True)
graph2 = pydotplus.graph_from_dot_data(dot_data)
Image(graph2.create_png())     # display on screen
graph2.write_pdf("myfile.pdf") # save to file
```

Python Example - Car Cylinder Discrimination (4/4)



Python Example - PCA (2)



References

1. MPG data from: Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository `http://archive.ics.uci.edu/ml`. Irvine, CA: University of California, School of Information and Computer Science.