Trees and Ensemble Methods

Vahid Partovi Nia

Lecture 08

Outline

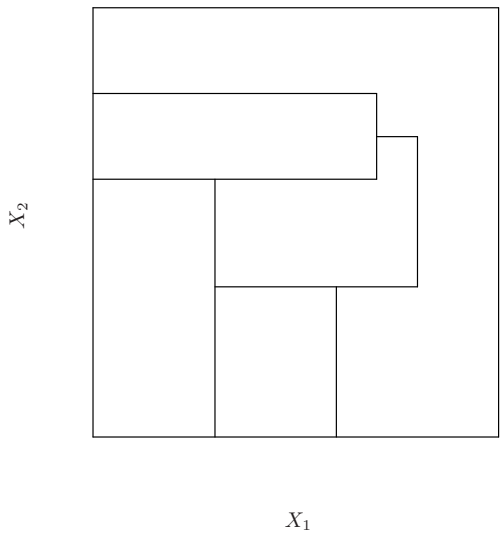
CART

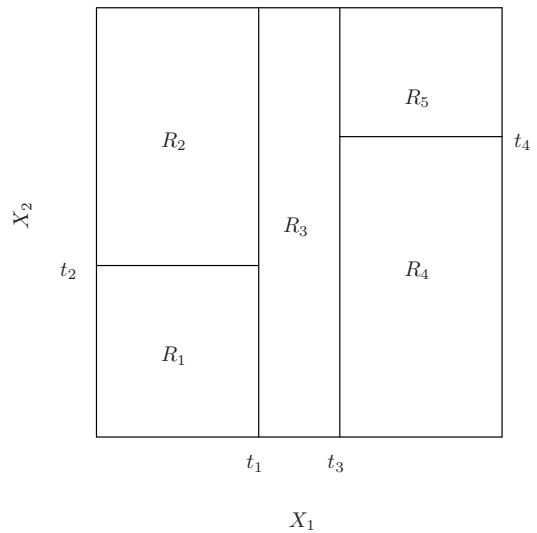
Bagging

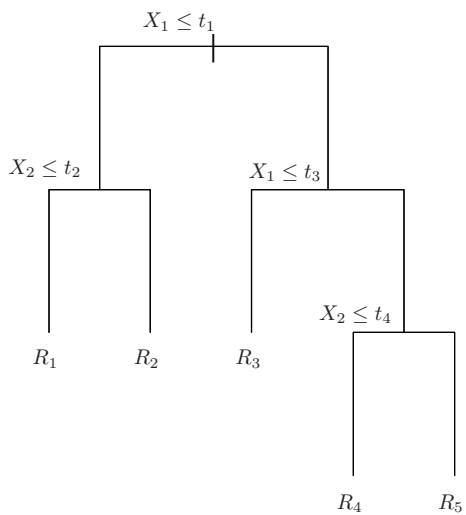
Boosting

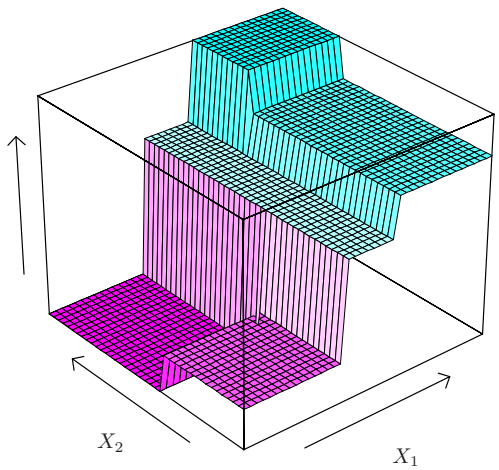
1. CART
2. Bagging
3. Boosting

Classification and Regression Trees

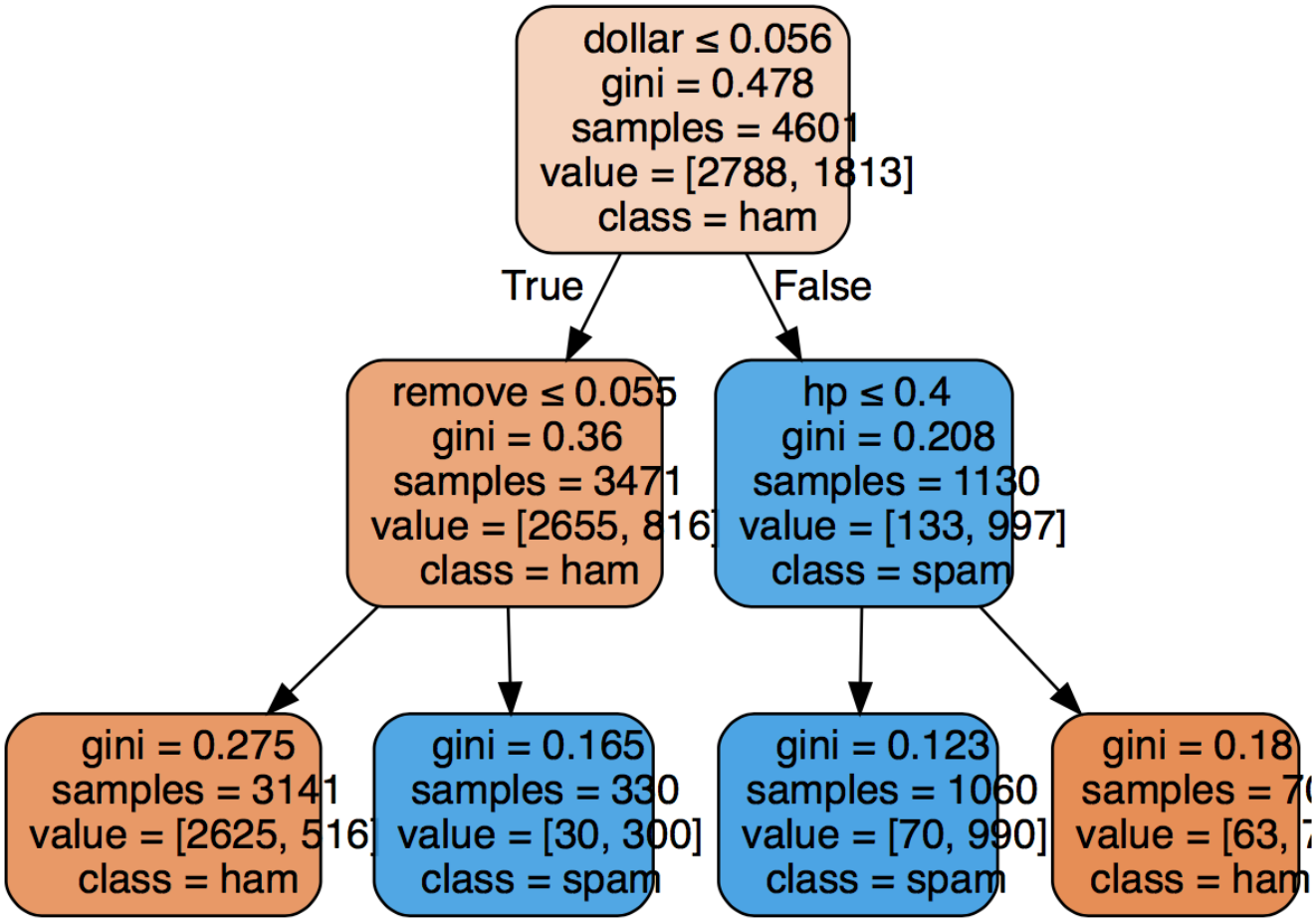






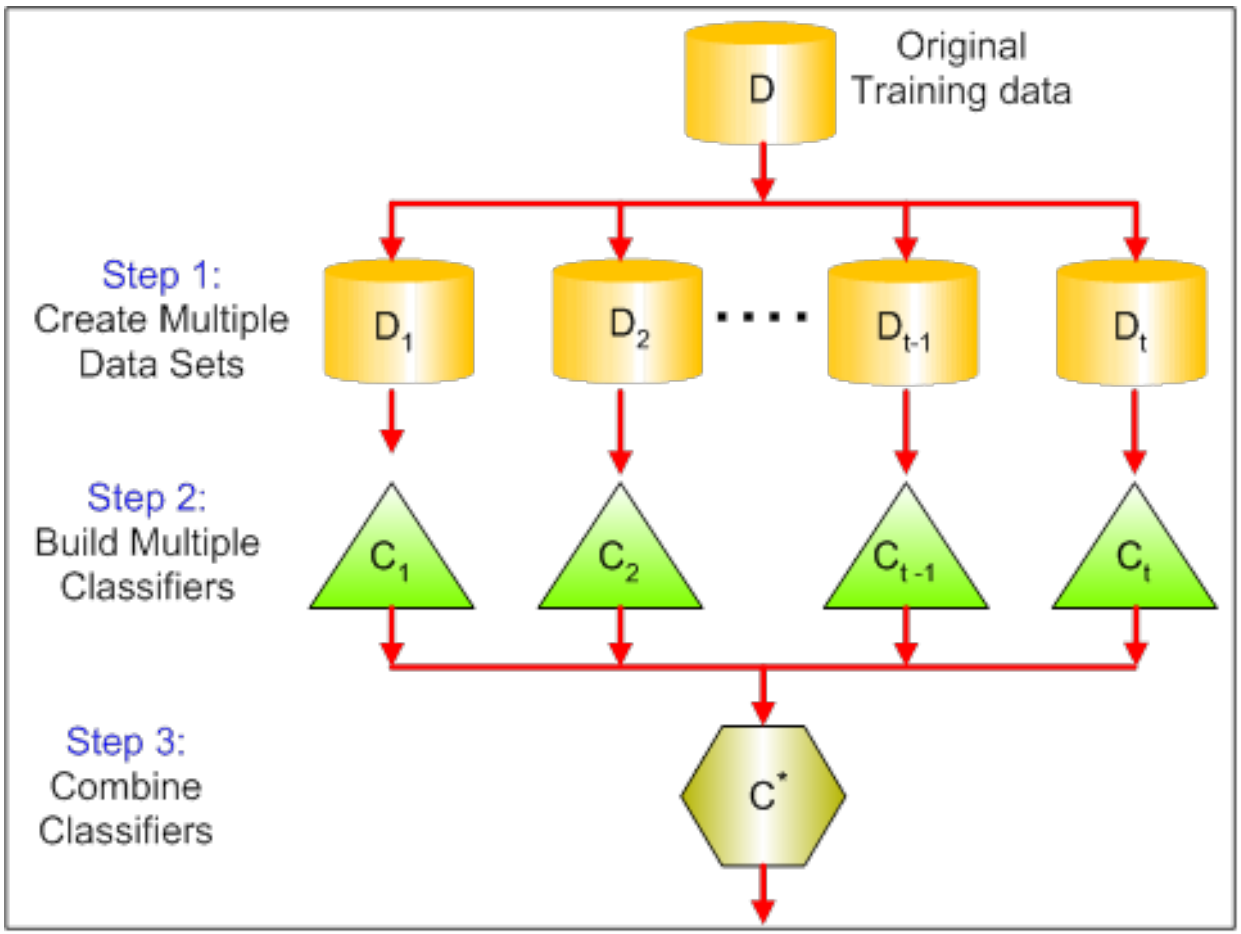


```
import pandas as pd
path='data/'
filename = path+'spamdata01.csv'
spam = pd.read_csv(filename)


import matplotlib.pyplot as plt
%matplotlib inline
plt.scatter(spam.values[:,11], spam.values[:,-1]);


from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz

dt = DecisionTreeClassifier(max_depth=3)

X = spam.values[:, :57]
y = spam.values[:, -1]
dt.fit(X,y)
spamnames = spam.columns.tolist()[:57]


dot_data = export_graphviz(dt, out_file=None,
                           feature_names=spamnames,
                           class_names=['ham', 'spam'],
                           filled=True, rounded=True,
                           special_characters=True)


import graphviz
graph = graphviz.Source(dot_data)
graph
```


dollar ≤ 0.056
gini = 0.478
samples = 4601
value = [2788, 1813]
class = ham
True
False
remove ≤ 0.055
gini = 0.36
samples = 3471
value = [2655, 816]
class = ham
hp ≤ 0.4
gini = 0.208
samples = 1130
value = [133, 997]
class = spam
gini = 0.275
samples = 3141
value = [2625, 516]
class = ham
gini = 0.165
samples = 330
value = [30, 300]
class = spam
gini = 0.123
samples = 1060
value = [70, 990]
class = spam
gini = 0.18
samples = 7
value = [63,
class = ham

```
dt10 = DecisionTreeClassifier(max_depth=10)
dt10.fit(X_train, y_train)
y10_pred = dt10.predict(X_test)


from sklearn.metrics import accuracy_score
accuracy_score(y_test, y10_pred)
```

Bagging


D
Original
Training data
Step 1:
Create Multiple
Data Sets
D1
D2
Dt-1
Dt
Step 2:
Build Multiple
Classifiers
C1
C2
Ct-1
Ct
Step 3:
Combine
Classifiers
C*

```
from sklearn.ensemble import BaggingClassifier
bag = BaggingClassifier(n_estimators=100, random_state=1)
bag.fit(X_train, y_train)
y_hat = bag.predict(X_test)
accuracy_score(y_test, y_hat)
```

http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html

```
from sklearn.ensemble import AdaBoostClassifier
adaboost = AdaBoostClassifier(n_estimators= 100, random_state=1)
adaboost.fit(X_train, y_train)
y_hat = adaboost.predict(X_test)
accuracy_score(y_test, y_hat)
```