9. Regression Trees

Bruce E. Shapiro

Getting Started in Machine Learning

Last revised: February 17, 2019

Regression Trees fit a Step Function

Replace the multi-linear model

$$y = a + b_0 x_0 + b_1 x_1 + \cdots + b_{n-1} x_{n-1}$$

with

$$\begin{aligned} y &= c_0 \delta(R_0, \mathbf{x}) + c_1 \delta(R_1, \mathbf{x}) + \cdots c_{K-1} \delta(R_{K-1}, \mathbf{x}) \\ &= \sum_{k=0}^{K-1} c_k \delta(R_k, \mathbf{x}) \end{aligned}$$

where $\{R_k\}_{k=0,1,\ldots}$ is a partition of the domain and

$$\delta(R_k, \mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in R_k \\ 0, & \text{otherwise} \end{cases}$$

- Objective function (K=num. partitions; N=num. of points)

$$\mathcal{E}_{\text{Regression Tree}} = \sum_{j=0}^{N-1} \sum_{i=0}^{K-1} (y_j - \hat{y}_i)^2 \delta(R_i, x_j)$$

- Algorithm identifies **cut points** that progressively subdivide sub-domains to minimize the objective function.

- Objective function (K=num. partitions; N=num. of points)

$$\mathcal{E}_{\text{Regression Tree}} = \sum_{j=0}^{N-1} \sum_{i=0}^{K-1} (y_j - \hat{y}_i)^2 \delta(R_i, x_j)$$

- Algorithm identifies **cut points** that progressively subdivide sub-domains to minimize the objective function.
- Sub-domains are split along a single feature axis.

- Objective function (K=num. partitions; N=num. of points)

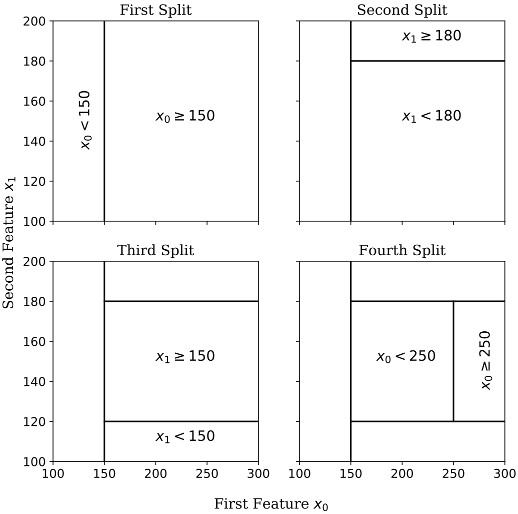
$$\mathcal{E}_{\text{Regression Tree}} = \sum_{j=0}^{N-1} \sum_{i=0}^{K-1} (y_j - \hat{y}_i)^2 \delta(R_i, x_j)$$

- Algorithm identifies **cut points** that progressively subdivide sub-domains to minimize the objective function.
- Sub-domains are split along a single feature axis.
- Average y value over each sub-domain is taken as predictor.

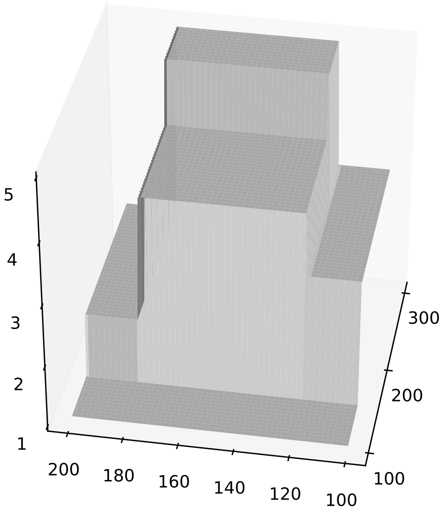
- Objective function (K=num. partitions; N=num. of points)

$$\mathcal{E}_{\text{Regression Tree}} = \sum_{j=0}^{N-1} \sum_{i=0}^{K-1} (y_j - \hat{y}_i)^2 \delta(R_i, x_j)$$

- Algorithm identifies **cut points** that progressively subdivide sub-domains to minimize the objective function.
- Sub-domains are split along a single feature axis.
- Average y value over each sub-domain is taken as predictor.
- Process *could* be repeated until there is only one point in each sub-domain (step-function-interpolation).

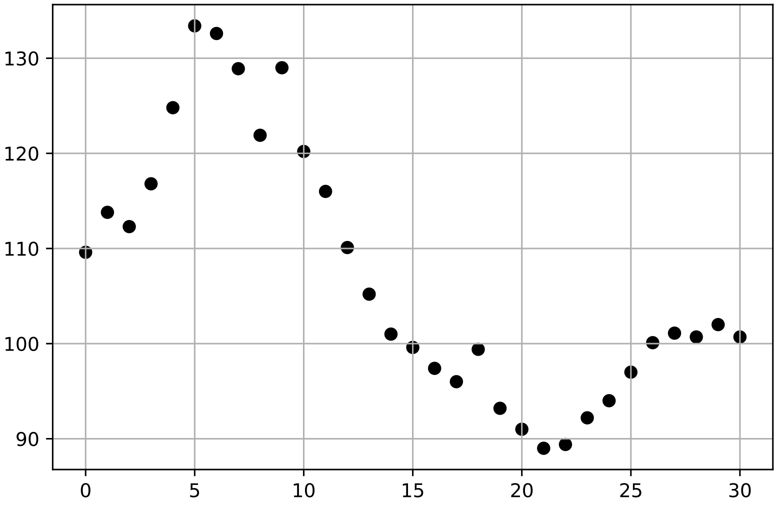


First Split
Second Split
Third Split
Fourth Split
$x_0 < 150$
$x_0 \geq 150$
$x_1 \geq 180$
$x_1 < 180$
$x_1 \geq 150$
$x_1 < 150$
$x_0 < 250$
$x_0 \geq 250$
100
120
140
160
180
200
100
150
200
250
300
Second Feature x_1
First Feature x_0

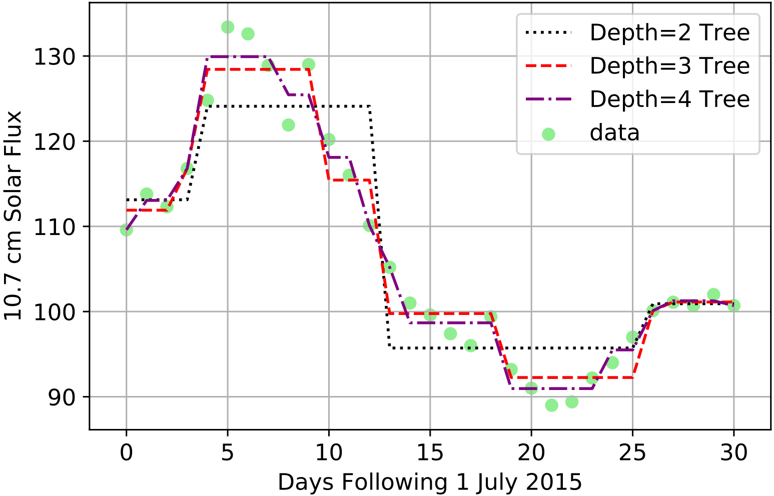

5
4
3
2
1
200
180
160
140
120
100
100
200
300

- **1D Example**. Solar Flux - July 2015

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

rawdata=pd.read_csv("solar-data-july-2015.txt",
   sep="\t",skiprows=15)
Y=(np.flip(np.array(rawdata["Flux"]))).reshape(-1,1)
n=len(Y)
X=np.array(list(range(n))).reshape(-1,1)
plt.scatter(X,Y)
```


130
120
110
100
90
0
5
10
15
20
25
30

```
from sklearn.tree import DecisionTreeRegressor

regr=DecisionTreeRegressor(max_depth=3)
regr.fit(X,Y)
YP=regr.predict(X)

plt.scatter(X,Y)
plt.plot(X,YP)
```

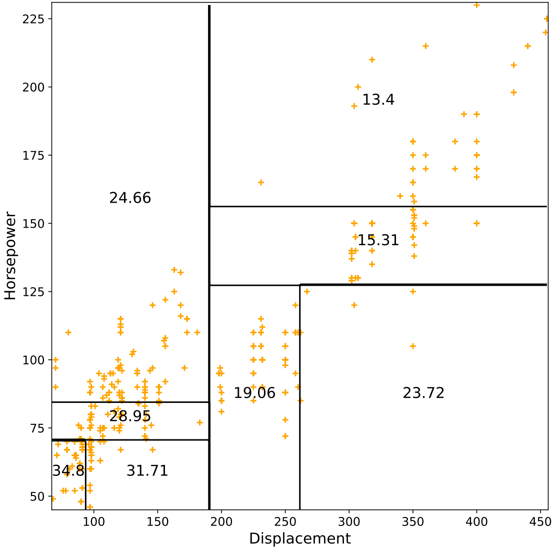

Depth=2 Tree
Depth=3 Tree
Depth=4 Tree
data
130
120
110
100
90
10.7 cm Solar Flux
0
5
10
15
20
25
30
Days Following 1 July 2015

```
from sklearn.metrics import mean_squared_error
for j in range(2,8):
    reg=DecisionTreeRegressor(max_depth=j)
    reg.fit(X,Y)
    YP=reg.predict(X)
    MSE=mean_squared_error(YP,Y)
    print("The MSE for a depth of ",j," is ",
      round(MSE,2))
```

```
The MSE for a depth of  2  is  25.71
The MSE for a depth of  3  is  8.32
The MSE for a depth of  4  is  3.73
The MSE for a depth of  5  is  0.81
The MSE for a depth of  6  is  0.11
The MSE for a depth of  7  is  0.0
```

Auto MPG Data

```
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)

X=np.array(data[["displ","hp"]])
Y=np.array(data["mpg"]).reshape(-1,1)
n=len(Y)

r=DecisionTreeRegressor(max_depth=3)
r.fit(X,Y)
YP=r.predict(X)
```


13.4
24.66
15.31
19.06
23.72
28.95
34.8
31.71
Horsepower
Displacement
225
200
175
150
125
100
75
50
100
150
200
250
300
350
400
450

Citations

1. Solar data from http://www.solen.info/solar/old_reports/2015/july/indices.html
2. Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth International Conference of Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann. (MPG Data). According the UCI website, “This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the 1983 American Statistical Association Exposition.”
3. 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.