STATS762 Regression for Data Science

Tree-based methods

May 30, 2019

Heart Disease

The data contains the information of 303 patients and the goal is to predict the presence of heart disease. [1]

[2]

[1] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[2] https://www.heartfoundation.org.nz

Heart Diseaser

age	age in years
sex	1=male, 0=female
cp	chest pain type;Value 1: typical angina – Value 2: atypical angina – Value 3: non-anginal pain – Value 4: asymptomatic
trestbps	resting blood pressure (in mm Hg on admission to the hospital)
chol	serum cholestoral in mg/dl
fbs	(fasting blood sugar $>$ 120 mg/dl) (1 = true; 0 = false)
restecg	resting electrocardiographic results
thalach	maximum heart rate achieved
exang	exercise induced angina (1 = yes; 0 = no)
oldpeak	ST depression induced by exercise relative to rest
slope	the slope of the peak exercise ST segment
ca	number of major vessels (0-3) colored by flourosopy
target	1=heart disease presence, 0=no presence

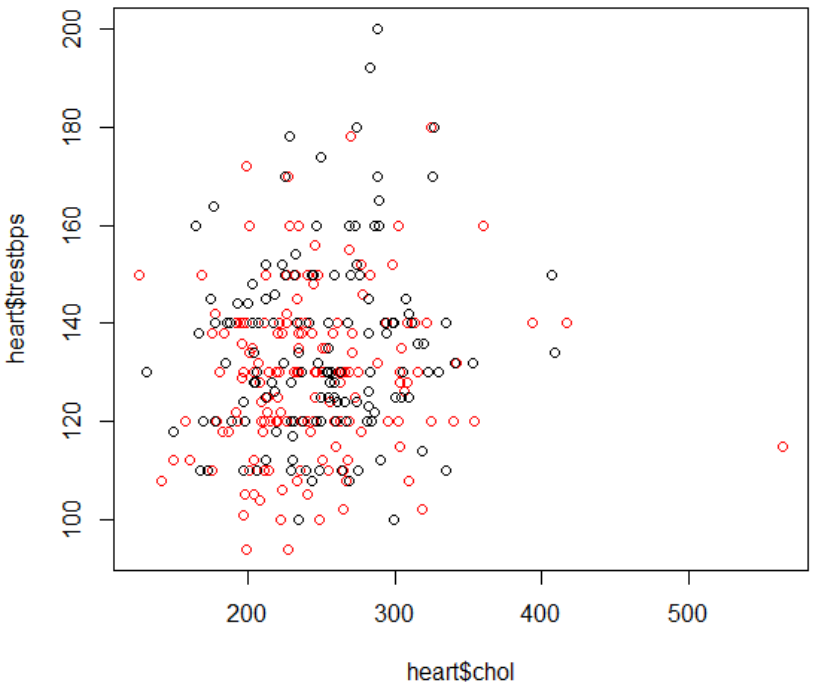
Heart Disease

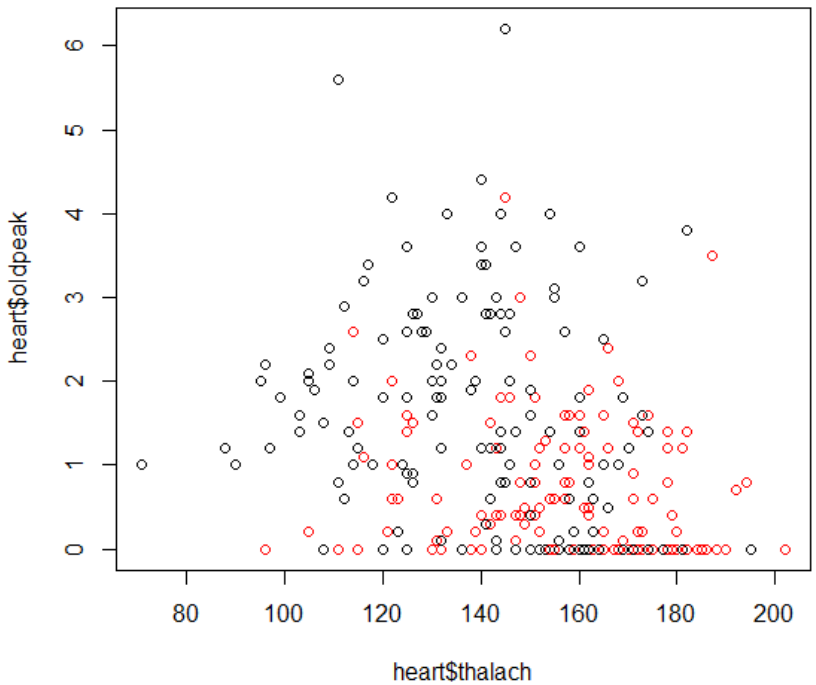
Since all entries are numbers, all variables are read as numeric variables. We need to specify nominal variables.

- continuous variable - age, trestbps, chol, trestbps, oldpeak
- numeric integer - restecg, slope, ca
  i.e., The order of value is meaningful.
- nominal variable - target, sex, cp, fbs, exang

```
> str(heart)
'data.frame':   303 obs. of  13 variables:
 $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
 $ sex     : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 2 2 ...
 $ cp      : Factor w/ 4 levels "0","1","2","3": 4 3 2 2 1 1 2 2 3 3 ...
 $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
 $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
 $ fbs     : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 2 1 ...
 $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
 $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
 $ exang   : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
 $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
 $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
 $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ target  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

Heart Disease





No presence (black dot) and heart disease presence (red dot)

Certainly boundaries are neither linear or quadratic forms.

Heart Disease

Freq table of six variables by target (0 or 1);

restecg	0	1	2
0	79	56	3
1	68	96	1

slope	0	1	2
0	12	91	35
1	9	49	107

sex	0	1
0	24	114
1	72	93

fbs	0	1
0	116	22
1	142	23

cp	0	1	2	3
0	104	9	18	7
1	39	41	69	16

exang	0	1
0	62	76
1	142	23

Different distribution against target - Useful variable to predict the target variable.
Similar distribution against target - Not useful variable to predict the target variable.

Motivation

We have some challenges in modelling classes;

- Mixed types of variables (numeric and categorical variables)
- Boundaries between classes are too complicated.
  i.e., higher order polynomial, discontinuous set, inconsistent variable contribution.

Discriminant analysis or multinomial logistic regression model are designed to model the log-odd ratio and predict a class membership for each datapoint. Boundaries are derived from the log-odd ratio.

Boundaries may be too complex for the multinomial regression and LDA/QDA.

Partition the feature space in to a set of rectangles and fit a simple model (like a constant) in each one.

”Tree-based methods”

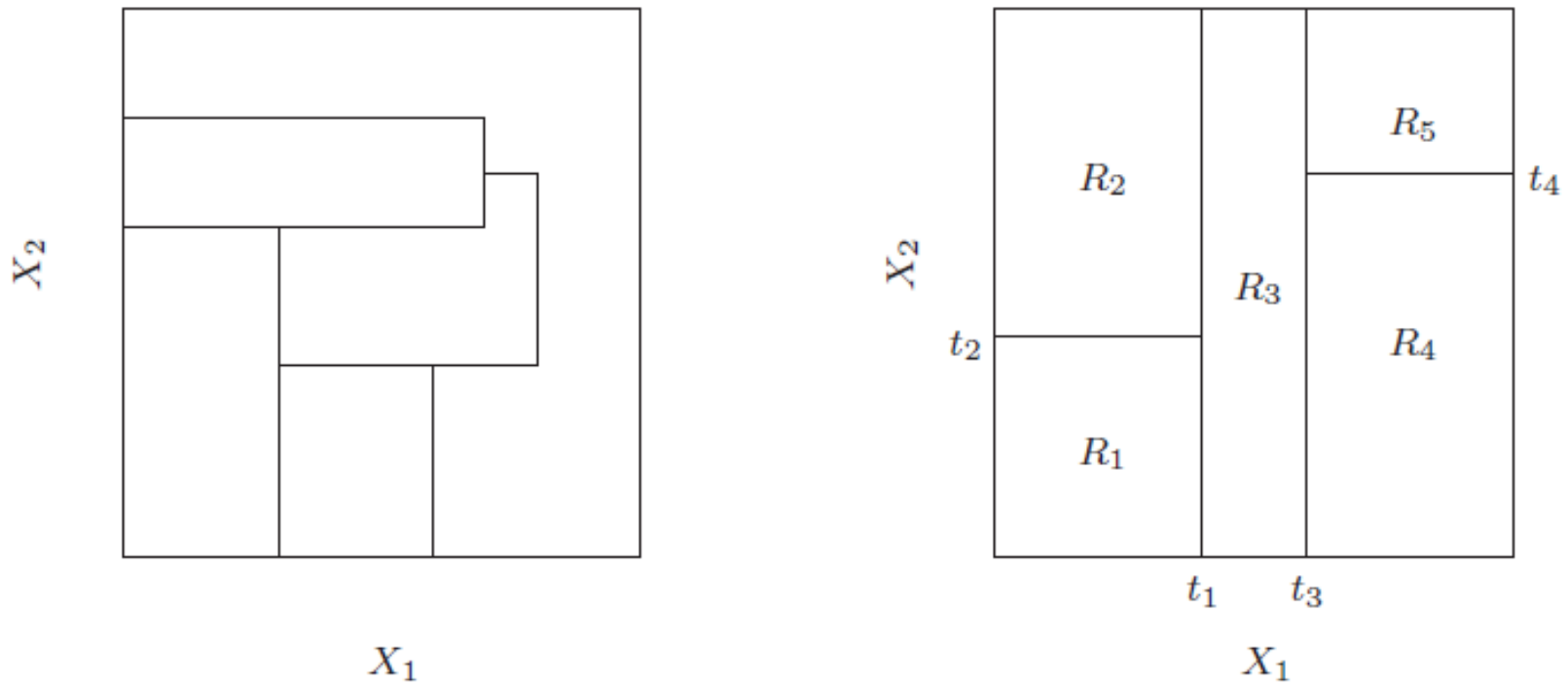
Classification And Regression Trees (CART)

Tree-Based methods

Recursive partitioning

- Consider a linear regression problem with a continuous response y and two predictors x_1 and x_2
- Split the space into two regions on the basis of a rule and modeling the response using the mean of y in the two regions.
- The optimal split (in terms of reducing the loss) is found over all variables j and all possible split points t.
- The process is then repeated in a recursive fashion for each of the two sub-regions.

Tree-Based methods



Right : Recursive partitions - Four splitting points (t_1-t_4) yield five sub-regions (R_1-R_5). [3]

[3] The Elements of Statistical Learning. Hastie, T., Tibshirani, R., Friedman, J. Spring Series in Statistics.

Tree-Based methods

- This process continues until some stopping rule is applied
- Let $\{R_m\}$ denote the collection for rectangular partitions, we might continue partitioning until $\{R_m\}$ reaches a priory set number of partitions, M.
- The end result is a piecewise constant model over the partition $\{R_m\}$ of the form

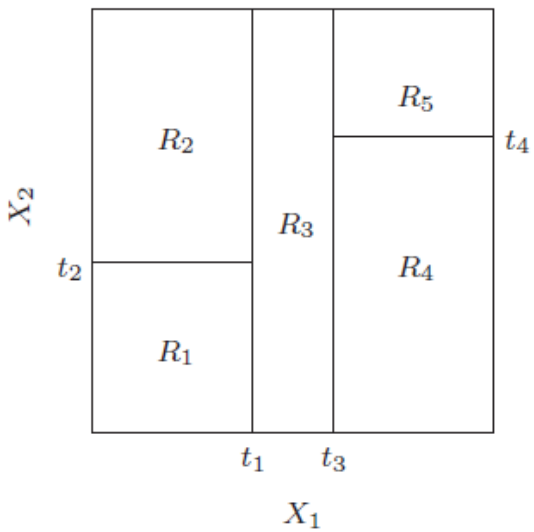
$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

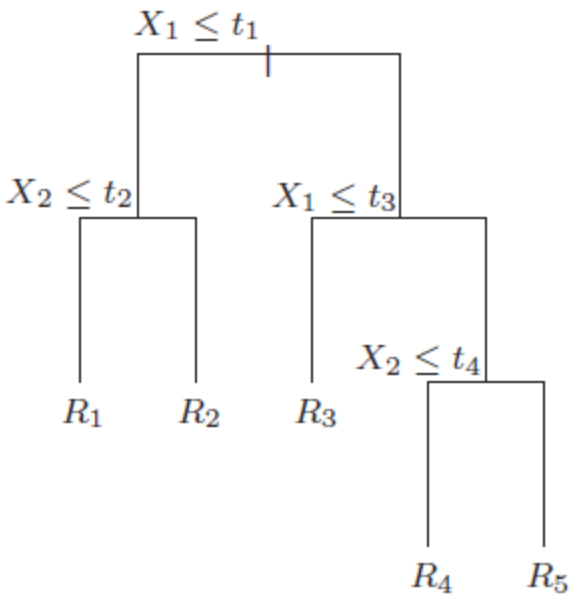
where c_m is the constant term for the m-th region (i.e., the mean of y_i for those observations $x_i \in R_m$)

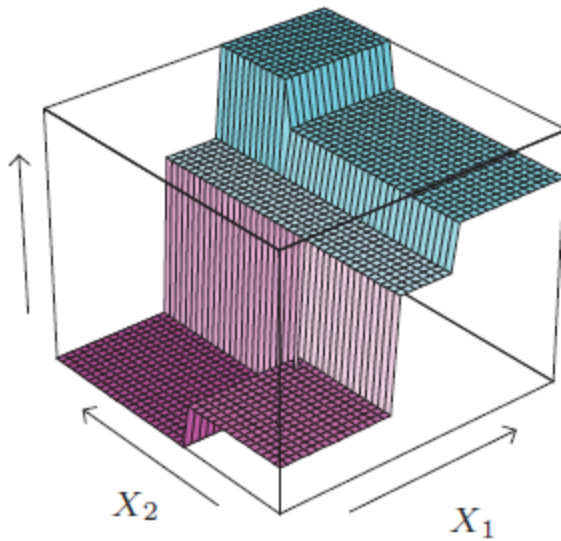
Tree-Based methods

- The same model can be neatly expressed in the form of a binary tree.
- The regions $\{R_m\}$ are then referred to as the terminal nodes of the tree
- The non-terminal nodes are referred to as interior nodes The splits are variously referred to as *splits*, *edges*, or *branches*

Tree-Based methods







4

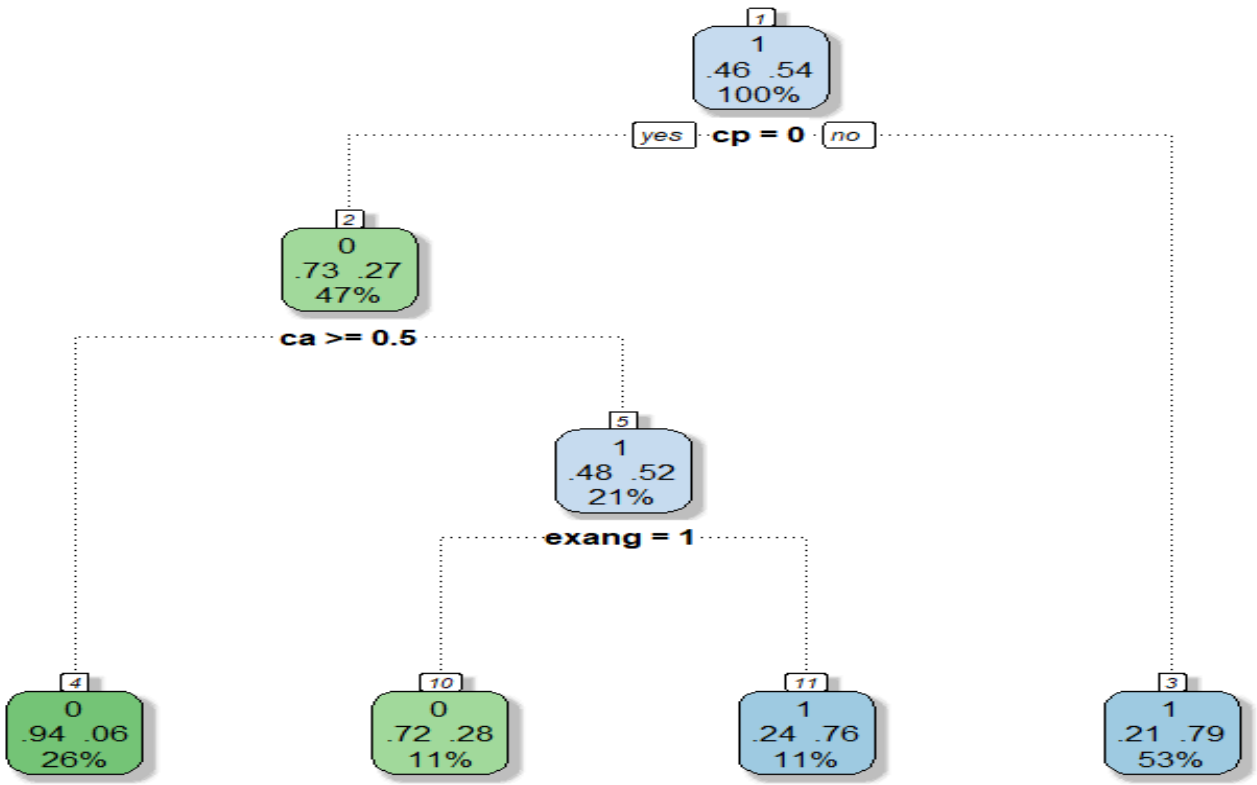
[4] The Elements of Statistical Learning. Hastie, T., Tibshirani, R., Friedman, J. Spring Series in Statistics.

Tree-Based methods

- Easy to implement and no assumption/restriction except the tree size
- Easy to interpret tree-representations.
- Applicable to more than two explanatory variables.
  The earlier partition diagram (square boxes) becomes difficult to draw, but the tree representation can be extended to any dimension.
- Popularly used in many applied areas

Tree-Based methods

Example : How can we predict the presence of heart disease?



Tree-Based methods

Terminologies

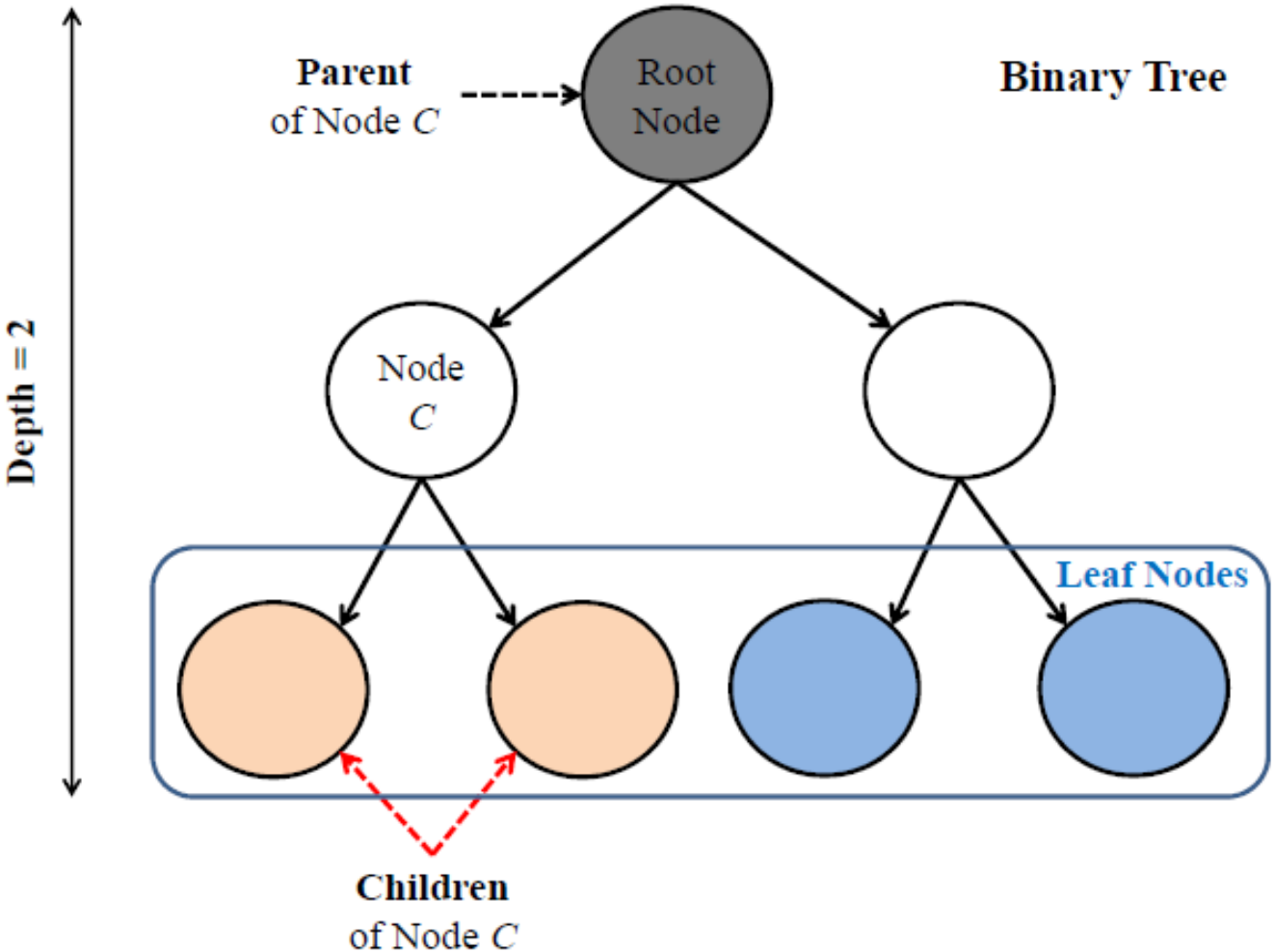
- The **nodes** of a decision tree can be classified in internal nodes and terminal nodes.

  Internal node : Decision-making unit that evaluates a decision function to determine which child node to visit next.

  Terminal node (leaf) : No child nodes and no associated with one of the partitions of the input space.

- **Root node** is the top node of the tree; the only node without parents.
- **Depth of a tree** is the maximal length of a path from the root node to a leaf node.

Tree-Based methods



Tree-Based methods

Given the data set $D = (x_1, y_1), \ldots, (x_N, y_N)$, the exploratory variable is $x_i = (x_{i1}, \ldots, x_{ip})$ and the response variable is y_i.

The algorithm partition the domain into M regions, $R_1, R_2, \ldots, R_M$ and the response in each region is modelled by c_m

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m),$$

and c_m is a value minimizing the impurity of R_m.

The splits should divide the observations within a node such that the impurity of each new partitions is minimized. i.e., Each partition, y's are very similar.

Classification Trees

Response variable is a nominal variable; $y_i \in \{1, 2, ..., K\}$.

Let N_m be a number of observations in R_m. Proportion of class k observations in node m is

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

where $I(y_i = k) = 1$ only if $y_i = k$. Otherwise $I(y_i = k) = 0$.
i.e., If all observations belong to the same class J in node m, $\hat{p}_{mJ} = 1$ and $\hat{p}_{mk} = 0$ for $k \neq J$.

Gini index is a popular measure of node impurity $Q_m(T)$

$$G(m) = \sum_{k=1}^{K} \hat{p}_{mk} \sum_{k' \neq k} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) = 1 - \sum_{k=1}^{K} \hat{p}_{mk}^2$$

and this index is used for splitting nodes and pruning nodes.

The best binary partition in terms of minimum impurity is with the highest probability in each region

$$\hat{c}_m = \text{argmax}_k \hat{p}_{mk}$$

Regression Trees

Response variable is a nominal variable; $y_i \in \mathbb{R}$.

Let N_m be a number of observations in R_m. The impurity of R_m is

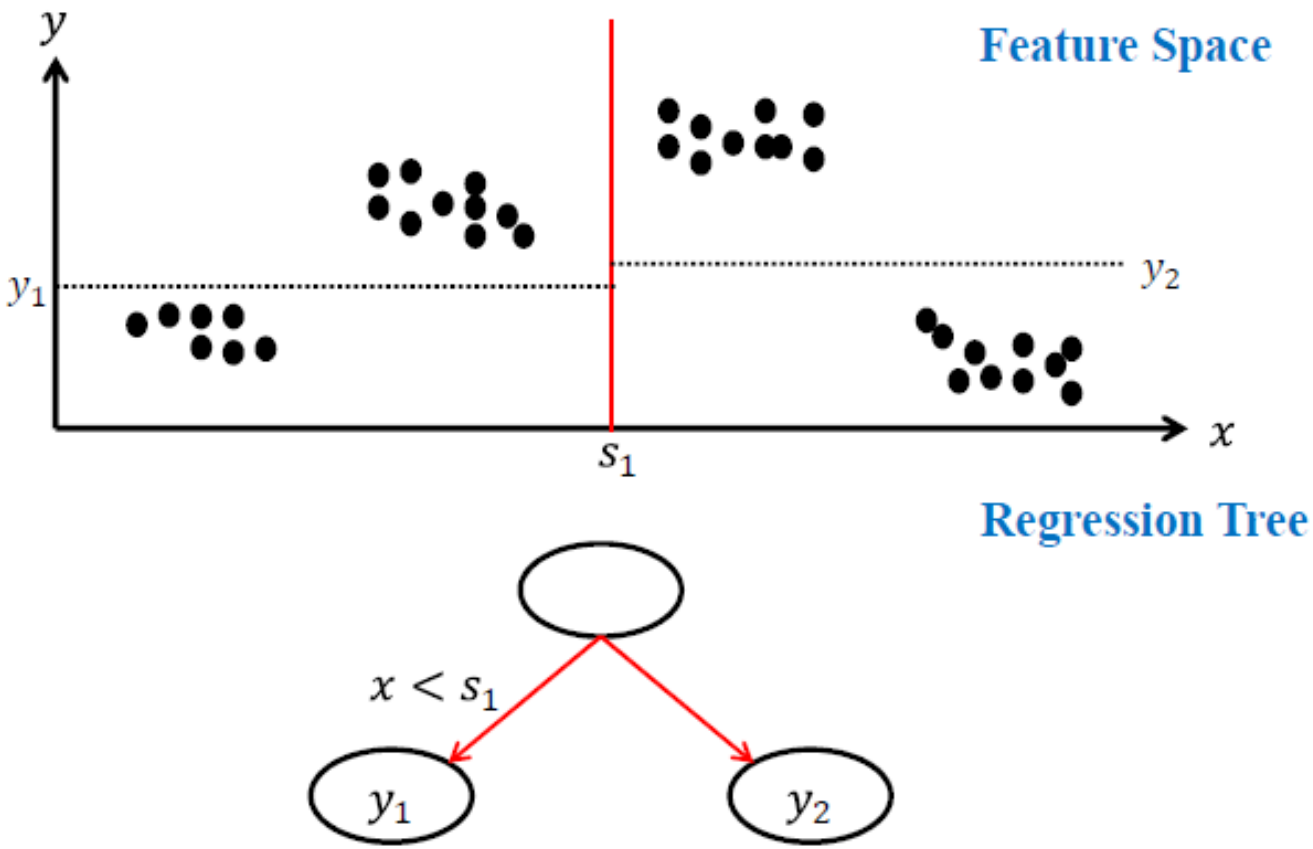
$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - c_m)^2$$

and c_m is a value minimizing Q_m

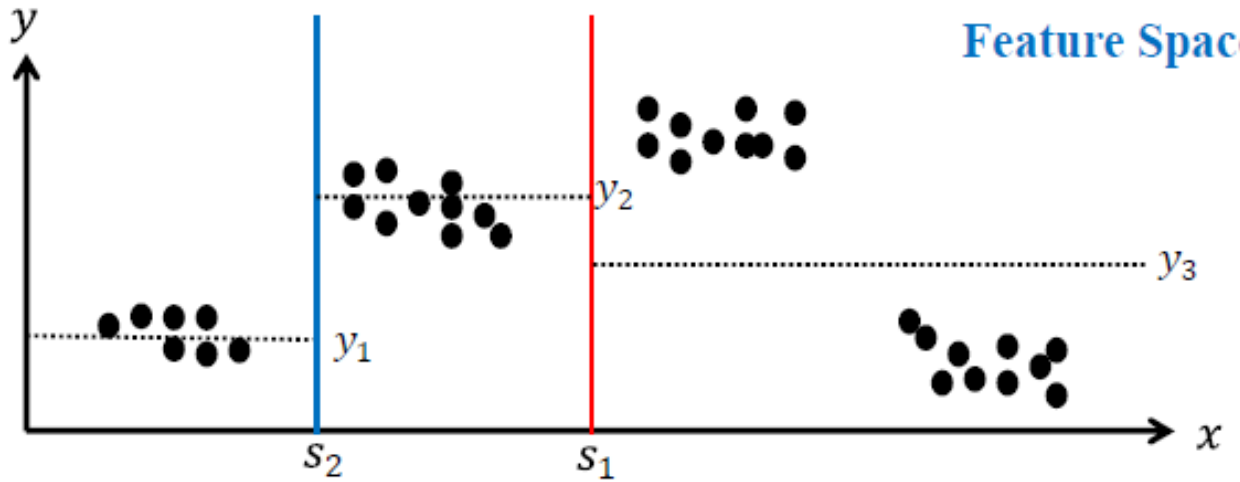
$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i \,.$$

i.e., If all observations are exactly equal in node m, $\hat{c}_m = y_i$ and $Q_m(T) = 0$.

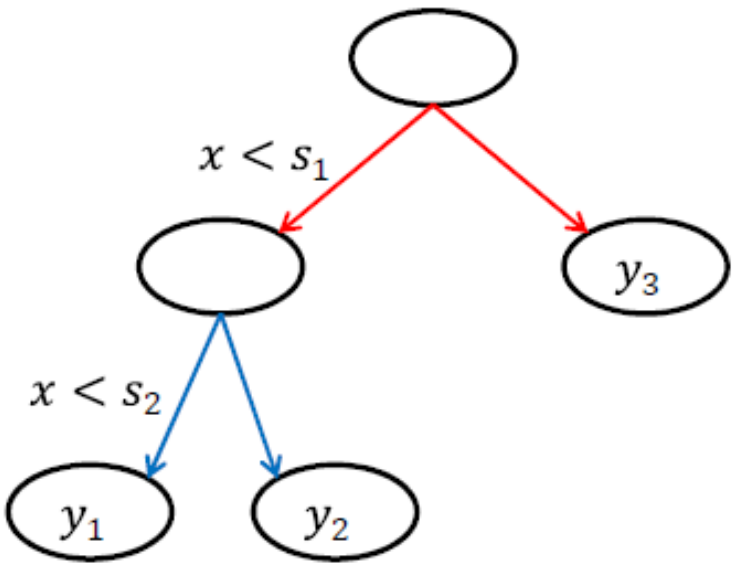
Tree-Based methods



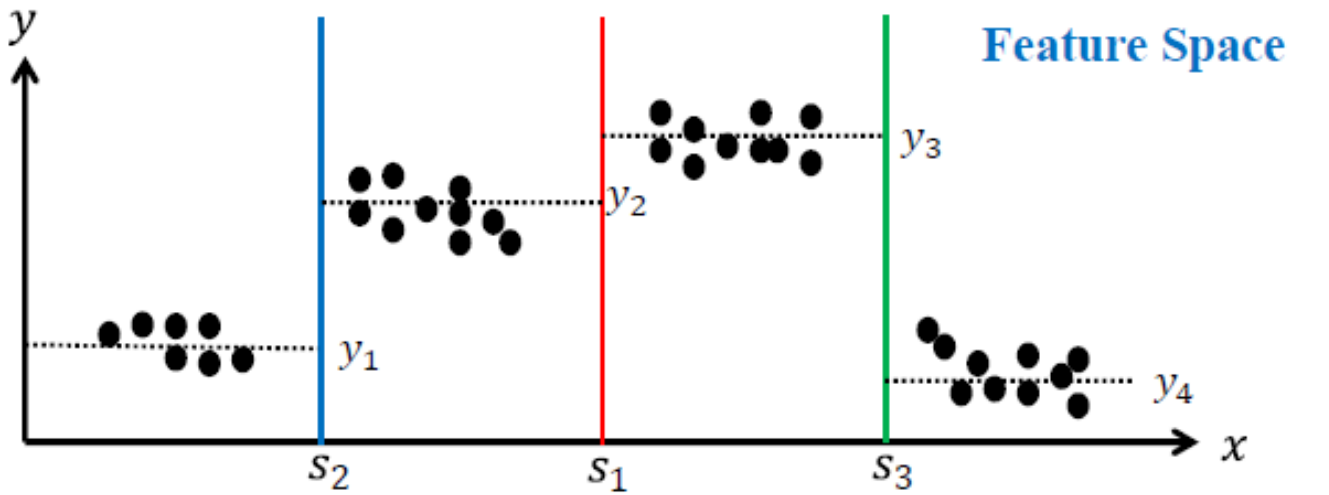
Tree-Based methods



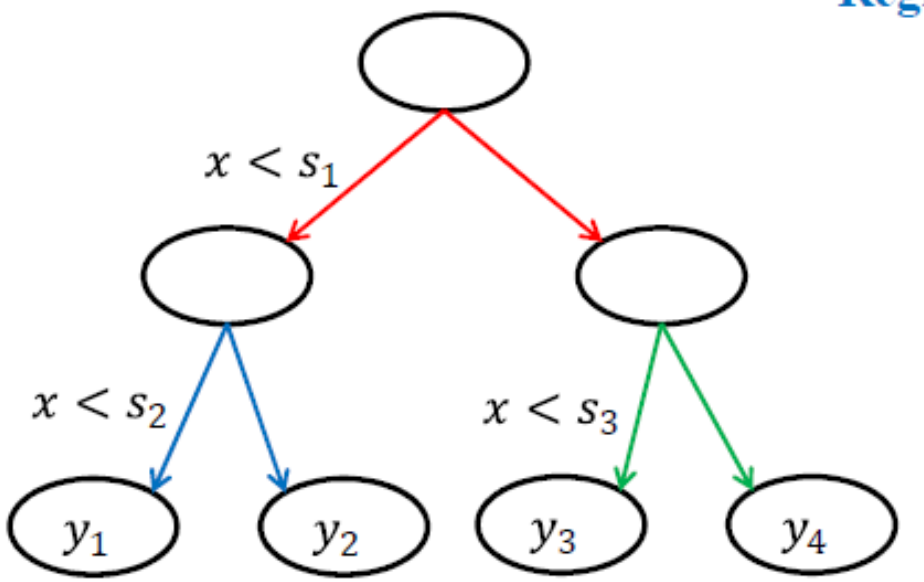
Regression Tree



Tree-Based methods



Regression Tree



Tree-Based methods

A splitting variable j and a split point s create the pair of regions

$$R_1(j,s) = \{X|X_j \leq s\} \quad \text{and} \quad R_2(j,s) = \{X|X_j > s\}$$

A splitting variable j and split point s are found by minimizing the impurity Q_m

$$\min_{j,s}\left[\min_{c_1} Q(x_i \in R_1(j,s)) + \min_{c_2} Q(x_i \in R_2(j,s))\right]$$

For any choice j and s, $\hat{c}_1$ and $\hat{c}_2$ are found by minimizing $Q(x_i \in R_1(j,s))$ and $Q(x_i \in R_2(j,s))$.

Scanning through all variables and domains, the best pair (j,s) is chosen. Repeat the splitting process on each partitioned regions.

Algorithm for Classification/Regression Trees

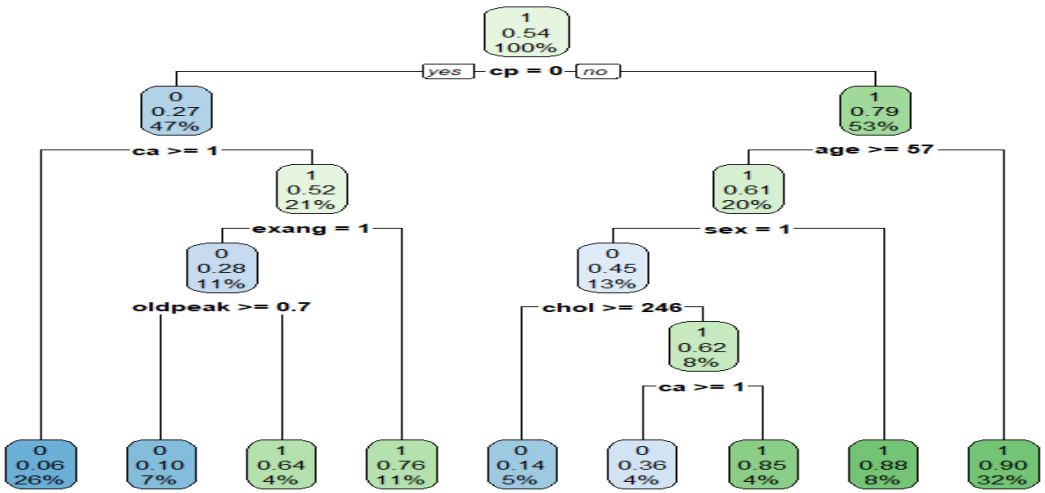
1. Start with $R_0 = \mathbb{R}^p$.
2. For each feature $j = 1, \ldots, p$, for each value $s \in \mathbb{R}$ we split regions:
   - 2.1 Split the data set
     $R_1 = \{x_i | x_{ij} \leq s\}$ and $R_2 = \{x_i | x_{ij} > s\}$
   - 2.2 Estimate parameters
     $\hat{c}_1 = \text{argmin}_c Q(x_i \in R_1)$ and $\hat{c}_2 = \text{argmin}_c Q(x_i \in R_2)$
   - 2.3 Quality of split is measured by the overall impurities

$$Q(x_i \in R_1(j, s)) + Q(x_i \in R_2(j, s))$$

3. Choose split (j, s) with minimal loss.
4. Repeat the splitting process on both regions until there is no reduction in loss by splitting regions.

Heart data

Largest classification tree - min overall impurity.




```
n= 303

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 303 138 1 (0.45544554 0.54455446)
   2) cp=0 143  39 0 (0.72727273 0.27272727)
     4) ca>=0.5 78   5 0 (0.93589744 0.06410256) *
     5) ca< 0.5 65  31 1 (0.47692308 0.52307692)
      10) exang=1 32   9 0 (0.71875000 0.28125000)
        20) oldpeak>=0.7 21   2 0 (0.90476190 0.09523810) *
        21) oldpeak< 0.7 11   4 1 (0.36363636 0.63636364) *
      11) exang=0 33   8 1 (0.24242424 0.75757576) *
   3) cp=1,2,3 160  34 1 (0.21250000 0.78750000)
     6) age>=56.5 62  24 1 (0.38709677 0.61290323)
      12) sex=1 38  17 0 (0.55263158 0.44736842)
        24) chol>=245.5 14   2 0 (0.85714286 0.14285714) *
        25) chol< 245.5 24   9 1 (0.37500000 0.62500000)
          50) ca>=0.5 11   4 0 (0.63636364 0.36363636) *
          51) ca< 0.5 13   2 1 (0.15384615 0.84615385) *
      13) sex=0 24   3 1 (0.12500000 0.87500000) *
     7) age< 56.5 98  10 1 (0.10204082 0.89795918) *
```

Tree-Based methods

How to find the optimal tree?

- Grow a large tree T_o stopping the splitting process only when some minimum node size is reached. Then this large tree is pruned using *cost-complexity pruning*.
- A subtree $T \subset T_o$ is obtained by pruning T_o, collapsing non-terminal nodes (leaf).

Tree-Based methods

Node m represents the region R_m and $|T|$ denotes the number of terminal nodes in T. Quantities for the region R_m are

- Number of datapoints in R_m : $N_m = \sum_i I(x_i \in R_m)$
- Representing value of R_m :

$$\hat{c}_m = \begin{cases} \frac{1}{N_m} \sum_{x_i \in R_m} y_i \,, & \text{continuous} \\ \text{argmax}_k \hat{p}_{mk} \,, & \text{nominal} \end{cases}$$

- Impurity measure in R_m :

$$Q_m(T) = \begin{cases} \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2 \,, & \text{continuous} \\ 1 - \sum_{k=1}^{K} \hat{p}_{mk}^2 \,, & \text{nominal} \end{cases}$$

Tree-Based methods

Cost complexity criterion is

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

where $\alpha \geq 0$. The subtree $T_\alpha \subset T_o$ is found minimizing the cost complexity criterion $C_\alpha(T)$.

Tuning parameter α is the tradeoff between tree size and its goodness of fit to the data.

- Large values of α result smaller trees T_α
- Small values of α result larger trees T_α
- $\alpha = 0$ results the full tree T_o

How to choose adaptively α?

Tree-Based methods

Weakest link pruning

- Collapse the internal node that produces the smallest per-node increase in $\sum_m N_m Q_m(T)$ and continue until we produce the single-node tree.
- This finite sequence of subtrees must contain the optimal tree T_α.
- Choose the value $\hat{\alpha}$ to minimize the cross-validated sum of squares and the final tree is T_α.
- Breiman et al. (1984) suggested that in actual practice, its common to instead use the smallest tree within 1 standard deviation of the minimum cross validation error (aka the 1-SE rule).

Classification/Regression tree in R

```
library(rpart)
rpart(formula, data, method, subset, parms, control)
```

- formula - regression formulae
- data - dataset
- method - type of responsible variable; one of ”anova”, ”poisson”, ”class” or ”exp”.
- subset - subset of the rows of the data should be used in the fit. (optional)
- parms - parameters for the splitting function (optional)
- control - a list of options that control details of the rpart algorithm

5

[5] https://cran.r-project.org/web/packages/rpart/rpart.pdf

Classification/Regression tree in R

Some control specifications and default values;

```
rpart.control(minsplit = 20, minbucket = round(minsplit/3),
cp = 0.01, maxcompete = 4, maxsurrogate = 5, xval = 10,
maxdepth = 30)
```

- minsplit - min number of observations for a splitting node
- minbucket - min number of observations in a terminal node (leaf)
- cp - complexity parameter for the largest tree. Any split that does not decrease the overall lack of fit by a factor of cp is not attempted.
- maxcompete - number of competitor splits retained in the output. Useful to diagnose variable importance.
- maxsurrogate - parameters for the splitting function (optional)
- xval - number of cross validations
- maxdepth - maximum depth of any node of the final tree

Heart data

Six α-values and corresponding classification trees

```
           CP nsplit rel error    xerror       xstd
1 0.471014493      0 1.0000000 1.0000000 0.06281757
2 0.061594203      1 0.5289855 0.5289855 0.05394174
3 0.024154589      3 0.4057971 0.4855072 0.05234715
4 0.021739130      6 0.3333333 0.4710145 0.05177794
5 0.003623188      8 0.2898551 0.4057971 0.04896008
6 0.001000000     10 0.2826087 0.3913043 0.04827210
```

- cp - complexity parameter α
- nsplit - number of node splits
- rel error - relative error, $1 - R^2$
- xerror - cross validation error
- xstd - cross validation standard deviation

Heart data

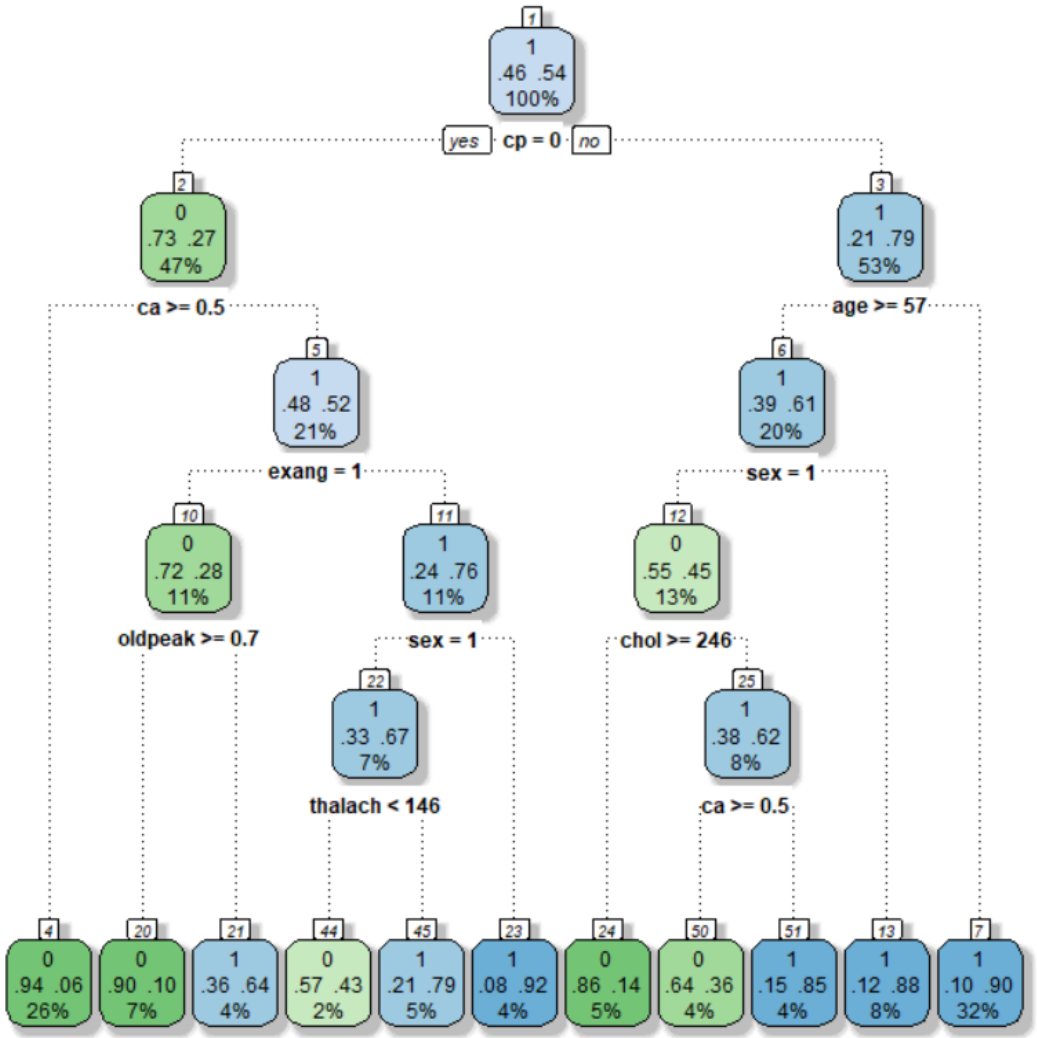
Six α-values and corresponding classification trees

	CP	nsplit	rel error	xerror	xstd
1	0.471014493	0	1.0000000	1.0000000	0.06281757
2	0.061594203	1	0.5289855	0.5289855	0.05394174
3	0.024154589	3	0.4057971	0.4855072	0.05234715
4	0.021739130	6	0.3333333	0.4710145	0.05177794
5	0.003623188	8	0.2898551	0.4057971	0.04896008
6	0.001000000	10	0.2826087	0.3913043	0.04827210

- When $\alpha = 0.01$, the cv-error is the minimum (0.001). The tree with the minimum loss.
- $\alpha = 0.003623188$ is the largest value in which the cv-error is within the 1sd around the min error (0.3913043 +0.04827210=0.4395764).

Heart data

Largest classification tree $\alpha = 0.001$



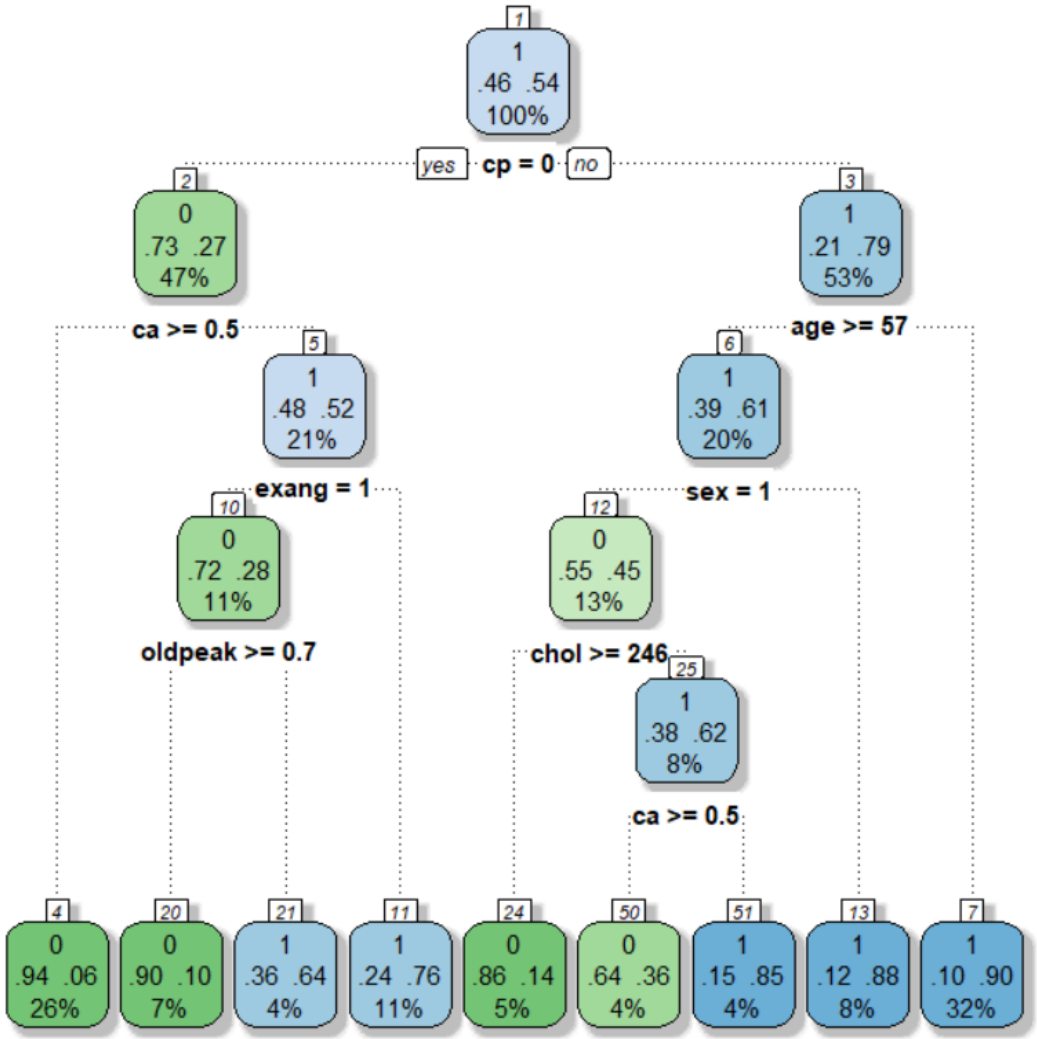

```
n= 303

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 303 138 1 (0.45544554 0.54455446)
   2) cp=0 143  39 0 (0.72727273 0.27272727)
     4) ca>=0.5 78   5 0 (0.93589744 0.06410256) *
     5) ca< 0.5 65  31 1 (0.47692308 0.52307692)
      10) exang=1 32   9 0 (0.71875000 0.28125000)
        20) oldpeak>=0.7 21   2 0 (0.90476190 0.09523810)
        21) oldpeak< 0.7 11   4 1 (0.36363636 0.63636364)
      11) exang=0 33   8 1 (0.24242424 0.75757576)
        22) sex=1 21   7 1 (0.33333333 0.66666667)
          44) thalach< 146 7   3 0 (0.57142857 0.42857143)
          45) thalach>=146 14   3 1 (0.21428571 0.78571429
        23) sex=0 12   1 1 (0.08333333 0.91666667) *
   3) cp=1,2,3 160  34 1 (0.21250000 0.78750000)
     6) age>=56.5 62  24 1 (0.38709677 0.61290323)
      12) sex=1 38  17 0 (0.55263158 0.44736842)
        24) chol>=245.5 14   2 0 (0.85714286 0.14285714) *
        25) chol< 245.5 24   9 1 (0.37500000 0.62500000)
          50) ca>=0.5 11   4 0 (0.63636364 0.36363636) *
          51) ca< 0.5 13   2 1 (0.15384615 0.84615385) *
      13) sex=0 24   3 1 (0.12500000 0.87500000) *
     7) age< 56.5 98  10 1 (0.10204082 0.89795918) *
```

Heart data

Optimal classification tree $\alpha = 0.003623188$ by the 1-SE rule.




```
n= 303

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 303 138 1 (0.45544554 0.54455446)
   2) cp=0 143  39 0 (0.72727273 0.27272727)
     4) ca>=0.5 78   5 0 (0.93589744 0.06410256) *
     5) ca< 0.5 65  31 1 (0.47692308 0.52307692)
      10) exang=1 32   9 0 (0.71875000 0.28125000)
        20) oldpeak>=0.7 21   2 0 (0.90476190 0.09523810)
        21) oldpeak< 0.7 11   4 1 (0.36363636 0.63636364)
      11) exang=0 33   8 1 (0.24242424 0.75757576) *
   3) cp=1,2,3 160  34 1 (0.21250000 0.78750000)
     6) age>=56.5 62  24 1 (0.38709677 0.61290323)
      12) sex=1 38  17 0 (0.55263158 0.44736842)
        24) chol>=245.5 14   2 0 (0.85714286 0.14285714)
        25) chol< 245.5 24   9 1 (0.37500000 0.62500000)
          50) ca>=0.5 11   4 0 (0.63636364 0.36363636) *
          51) ca< 0.5 13   2 1 (0.15384615 0.84615385) *
      13) sex=0 24   3 1 (0.12500000 0.87500000) *
     7) age< 56.5 98  10 1 (0.10204082 0.89795918) *
```

Heart data

Confusion matrices;

$\alpha = 0.003623188$	$y = 0$	$y = 1$
$y' = 0$	111	13
$y' = 1$	27	152

$\alpha = 0.001$	$y = 0$	$y = 1$
$y' = 0$	115	16
$y' = 1$	23	149

Overall accuracy rates are 0.867 for the optimal tree and 0.871 for the full tree.

Heart data

Relative variable importance : Relative improvement measures that each variable contributes as surrogate or splitter. A higher value means the more the variable contributes to improving the model.

```
        cp          ca     thalach       exang
0.208987671 0.137573203 0.136781402 0.127646298
   oldpeak         age       slope        chol
0.107676563 0.077262626 0.065952336 0.052014136
       sex    trestbps     restecg
0.047893790 0.035577678 0.002634295
```

Importance order of 12 variables is *cp* > *ca* > ... > *trestbps* > *restecg*.

Ozone data

The data contains 179 measurements of ozone concentration in the atmosphere. 7 main effects which, jointly with the quadratic terms and second order interactions, produce the above-mentioned $p = 35$ possible regressors.[6]

- y Response = Daily maximum 1-hour-average ozone reading (ppm) at Upland, CA
- x4 500-millibar pressure height (m) measured at Vandenberg AFB
- x5 Wind speed (mph) at Los Angeles International Airport (LAX)
- x6 Humidity (percentage) at LAX
- x7 Temperature (Fahrenheit degrees) measured at Sandburg, CA
- x8 Inversion base height (feet) at LAX
- x9 Pressure gradient (mm Hg) from LAX to Daggett, CA
- x10 Visibility (miles) measured at LAX

The rest 27 covariates are quadratic terms. For example, x4.x5=x4*x5

[6] Berger, J. and Molina, G. (2005) Posterior model probabilities via path-based pairwise priors. Statistica Neerlandica, 59:3-15.

Ozone data

Six α-values and corresponding regression trees

```
          CP nsplit rel error    xerror       xstd
1 0.61866992      0 1.0000000 1.0176762 0.10481017
2 0.04629404      1 0.3813301 0.4480629 0.05688866
3 0.04571914      2 0.3350360 0.4405023 0.05558595
4 0.03244033      3 0.2893169 0.4269072 0.05147647
5 0.03023071      4 0.2568766 0.4228301 0.05179909
6 0.01000000      5 0.2266459 0.3612418 0.04292384
```

When $\alpha = 0.01$, the cv-error is the minimum and this is also the optimal tree because there is no α within the 1se around the min cv error (0.3612418+0.04292384=0.4041656).

With the default value 0.01 may not be small enough to produce a deep tree and obtain the optimal tree.

Ozone data

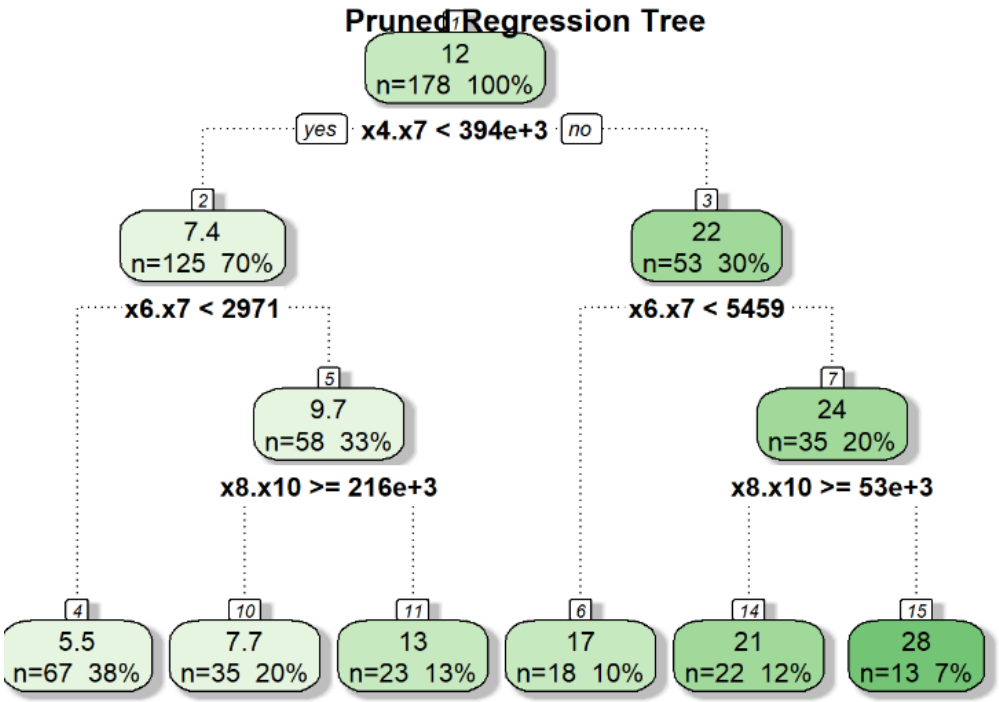
13 α-values and corresponding regression trees

	CP	nsplit	rel error	xerror	xstd
1	0.618669923	0	1.0000000	1.0084080	0.10379506
2	0.046294040	1	0.3813301	0.4577393	0.05818802
3	0.045719140	2	0.3350360	0.4602277	0.05528361
4	0.032440333	3	0.2893169	0.4591868	0.05340488
5	0.030230710	4	0.2568766	0.4254356	0.05163104
6	0.009754189	5	0.2266459	0.3945892	0.04728838
7	0.008480766	6	0.2168917	0.3755915	0.04713534
8	0.007594516	7	0.2084109	0.3781949	0.04704452
9	0.005404909	8	0.2008164	0.3794606	0.04717505
10	0.005390880	9	0.1954115	0.3740195	0.04481157
11	0.004129100	10	0.1900206	0.3680949	0.04403597
12	0.001518165	12	0.1817624	0.3691767	0.04329016
13	0.001000000	13	0.1802442	0.3654902	0.04337783

When $\alpha = 0.001000000$, the cv-error is minimized.
$\alpha = 0.009754189$ is the largest value in which the corresponding cv-error is within the 1sd around the min error (0.3654902 +0.04337783=0.408868)

Ozone data

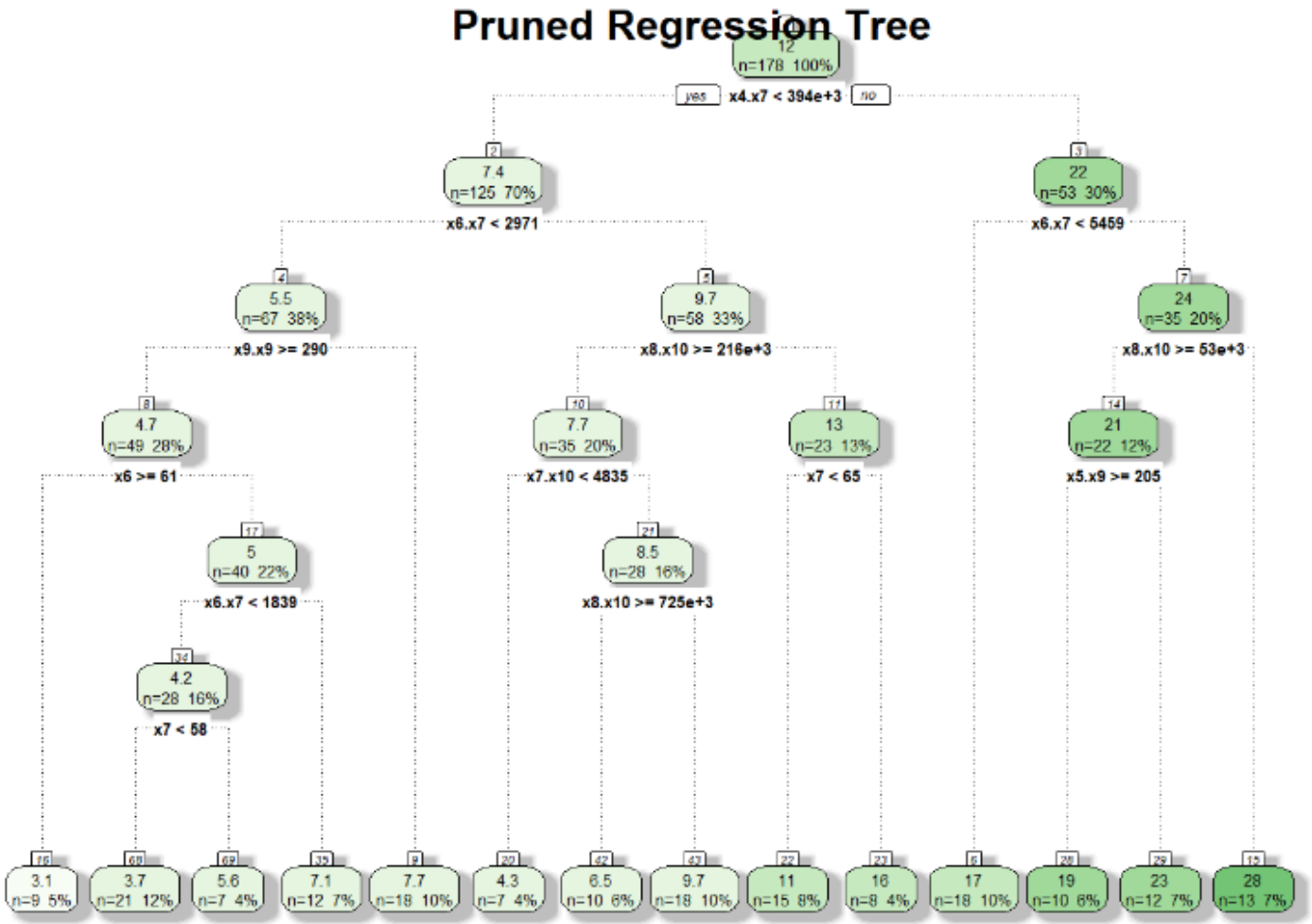
Regression tree with $\alpha = 0.009754189$




```
## n= 178
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 178 11926.9900 11.640450
##    2) x4.x7< 393990 125  2410.9120  7.448000
##      4) x6.x7< 2971 67   742.7463  5.492537 *
##      5) x6.x7>=2971 58  1116.0170  9.706897
##       10) x8.x10>=216190 35   379.5429  7.685714 *
##       11) x8.x10< 216190 23   375.9130 12.782610 *
##    3) x4.x7>=393990 53  2137.2080 21.528300
##      6) x6.x7< 5459 18   420.9444 17.055560 *
##      7) x6.x7>=5459 35  1170.9710 23.828570
##       14) x8.x10>=53165 22   530.3636 21.272730 *
##       15) x8.x10< 53165 13   253.6923 28.153850 *
```

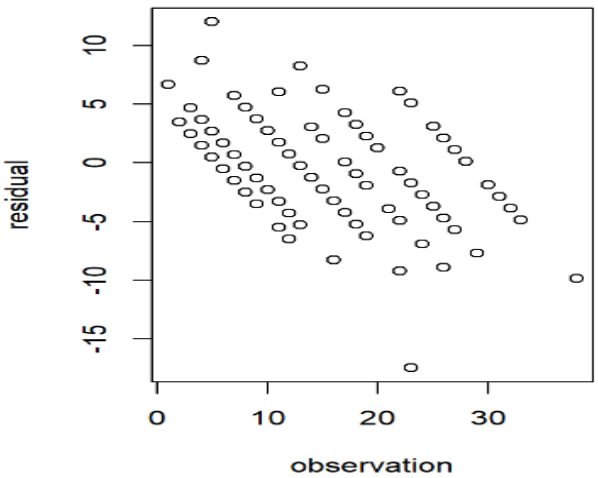
Ozone data

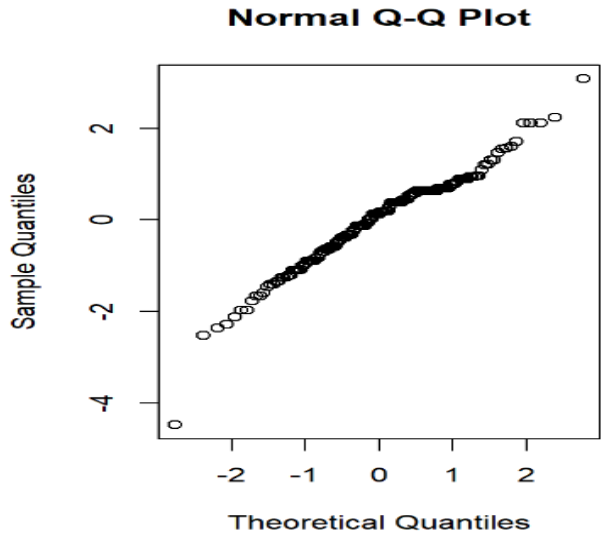
Regression tree with $\alpha = 0.0.001$



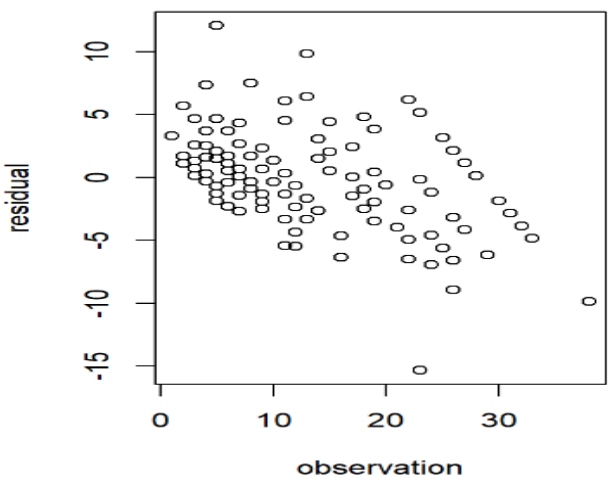
Ozone data

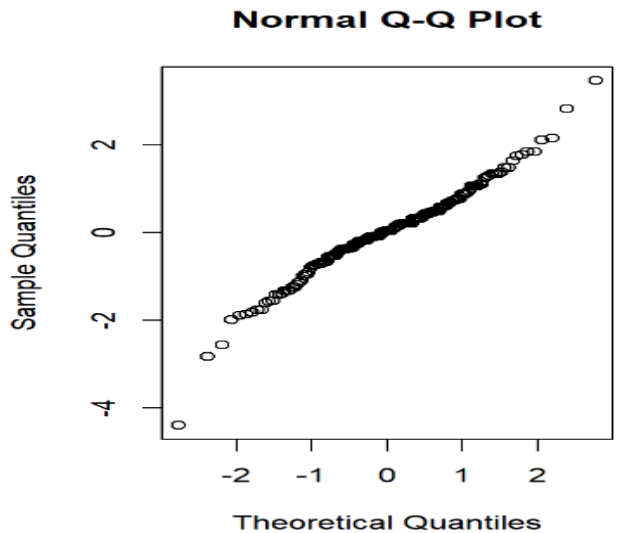
Regression tree with $\alpha = 0.009754189$ (top) and $\alpha = 0.001$ (bottom)





RMSE = 15.18653





RMSE = 12.07736

Ozone data

Relative variable importance

x4.x7	x7	x7.x7	x6.x7	x4
0.1727390134	0.1695943437	0.1592761043	0.1427625097	0.1034363679
x4.x4	x8.x10	x4.x6	x6	x6.x6
0.1034363679	0.0175680308	0.0167309156	0.0160153437	0.0152803887
x4.x8	x8	x6.x10	x4.x10	x6.x8
0.0121742118	0.0118945211	0.0073590844	0.0073074524	0.0067952168
x8.x8	x10	x5.x8	x9.x9	x4.x9
0.0067426938	0.0067151060	0.0054271254	0.0034910552	0.0027166003
x9	x7.x10	x10.x10	x5.x9	x5.x10
0.0027166003	0.0021885901	0.0015632787	0.0013911984	0.0012506229
x9.x10	x7.x8	x6.x9	x5.x7	
0.0011129587	0.0010607686	0.0009738389	0.0002796907	

Importance order of 35 variables is $x4.x7 > x7 > ... > x7.x8 > x5.x7$

Tree-Based methods

- Tree-based methods are not really statistical models - there is no distribution, no likelihood, no statistics we usually associate with modelling.
- Based on more algorithmic and treat the mechanism by which the data were generated as unknown.
- Easy to use and easy interpretation.

Problems and concerns :

- No theories to explain the underlying behaviour and characters of the data.
- Heuristic training techniques.
- Finding partition of space that minimizes empirical error is hard.

Gradient boosting tree

Motivation is to improve the tree method by reducing loss more effectively.

Gradient Boosting = Gradient Descent + Boosting

Boosting methods

Originally designed for classification problems then extended to regression.

Procedure combining the weak classifiers to produce a powerful one.

Adaboost

- Firstly we apply equal weights to each of the observations (x_i, y_i), $i = 1, ..., n$ and find a classifier. (standard classification tree)
- At each iteration, the observation weights are individually modified and the classification algorithm is reapplied to weighted observations.

  - Weights on observations misclassified by the previous classifier increase.
  - Weights on observations correctly classified by the previous classifier decrease.m
- Each classifier is forced to concentrate to those observations misclassified by the previous classifier.

Boosting methods

Notation

Given a tree with $R_1, ..., R_M$ regions, $x_i \in R_m$ implies that $f(x_i) = c_m$. The tree is formally expressed

$$T(x; \Theta) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

with parameters $\Theta = \{R_m, c_m\}_{m=1}^{M}$. Those parameters are found by minizing the overall loss

$$\hat{\Theta} = \text{argmin}_{\Theta} \sum_{m=1}^{M} \sum_{x_i \in R_m} L(y_i, c_m)$$

where L is a loss function.

- Numeric y - absolute error or squared error or Huber error
  In this class, we will use the squared error for L.
- Nominal y - multinomial deviance loss function

Boosting methods

Known R_m c_m is easily found minimizing the overall loss in R_m; $\hat{c}_m = \text{argmin}_{c_m} \sum_{x_i \in R_m} L(y_i, c_m)$.

Finding R_m Both R_m and c_m are found sequentially. Top-down recursive partitioning approach, R_m is found by

$$\hat{\Theta} = \text{argmin}_{\Theta} \sum_{i=1}^{n} \bar{L}(y_i, T(x_i, \Theta))$$

then given R_m, find c_m.

Boosting methods

The boosted tree model is a sum of trees

$$f_J(x) = \sum_{j=1}^{J} T(x; \Theta_j)$$

and at each step, the j-th tree is obtained by

$$\hat{\Theta}_j = \text{argmin}_{\Theta_m} \sum_{i=1}^{n} L(y_i, f_{j-1}(x_i) + T(x_i; \Theta_j))$$

where $\Theta_j = \{R_{jm}, c_{jm}\}_{m=1}^{M_j}$ and $f_{j-1}(x_i)$ is a tree output of x_i using $T(x; \Theta_{j-1})$.

i.e., $T(x; \Theta_j)$ is a tree fitting for error of $T(x; \Theta_{j-1})$.

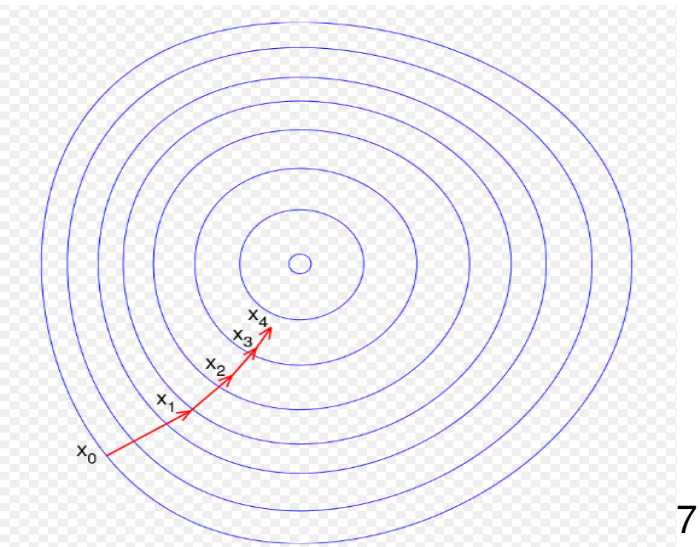
Difficult part is finding R_{jm} for $T(x; \Theta_j)$!

Gradient boosting tree

Revision - Gradient descent method
$F(x)$ decreases faster if one goes from x in the direction of the negative gradient of F at x_t – $F'(x_t)$. This follows

$$x_{t+1} = x_t - \gamma F'(x_t)$$

where γ is a step size.



[7]

[7] https://en.wikipedia.org/wiki/Gradient_descent

Gradient boosting tree

Let's find the solution tree minimizing L given the current tree and its fit using the gradient descent method.

The loss in using $f(x)$ to predict y is

$$L(f) = \sum_{i=1}^{n} L(y_i, f(x_i))$$

The goal is to minimize L with respect to the current tree f_{j-1}

$$g_{ij} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x_i)=f_{j-1}(x_i)}$$

and the solution is updated by $f_j = f_{j-1} + \gamma g_j$.

$L(f)$ is rapidly decreasing at f_{j-1} and as repeating this step, the final $L(f)$ approaches to the minimum.

Gradient boosting tree - Algorithm

1. Initialize $f_0(x) = \text{argmin}_\gamma \sum_{i=1}^{n} L(y_i, \gamma)$.
2. For $j = 1, 2, ..., J$
   - 2.1 For $i = 1, 2, ..., n$ compute

$$r_{ij} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{j-1}}$$

   - 2.2 Fit a regression tree to the targets r_{ij} giving terminal regions R_{jm}, $m = 1, 2, ..., M_j$.
   - 2.3 For $m = 1, 2, ..., M_j$ compute

$$\gamma_{jm} = \text{argmin}_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{j-1}(x_i) + \gamma)$$

   - 2.4 Update $f_j(x) = f_{j-1}(x) + \sum_{m=1}^{M_j} \gamma_{jm} I(x \in R_{jm})$
3. Output $\hat{f}(x) = f_J(x)$.

Gradient boosting tree

If each tree fits the train data too well...

- overfitting
- degrade the risk on future prediction
- small number of iterations (small number of trees) - L at the final model closes to the minimum after a short number of iterations.

If each tree fits the train data poorly...

- underfitting
- increase the risk on future prediction
- large number of iterations (many trees) - L at the final model closes to the minimum after long iterations.

Gradient boosting tree

Optimal tree size M^* is obtained by minimizing the future risk.

Simply scale the contribution of each tree by a factor $0 < \nu < 1$ when it is added to the current approximation

$$f_j(x) = f_{j-1}(x) + \nu \sum_{m=1}^{M_j} \gamma_{jm} I(x \in R_{jm}) .$$

ν is called a learning rate of the boosting procedure.

Small ν results small learning, better test error and larger training risk for a given J.

Friedman (2001)[8] recommended to use a small value for ν ($\nu < 0.1$) then choose J by early stopping.

[8] Greedy function approximation: A gradient boosting machine. (2001) Friedman, J. H. The Annals of Statistics. 29 (5) 1185-1232

Gradient Boosted in R

```
library(gbm)
gbm(formula, data, distribution, n.trees = 100,
shrinkage = 0.1, cv.folds)
```

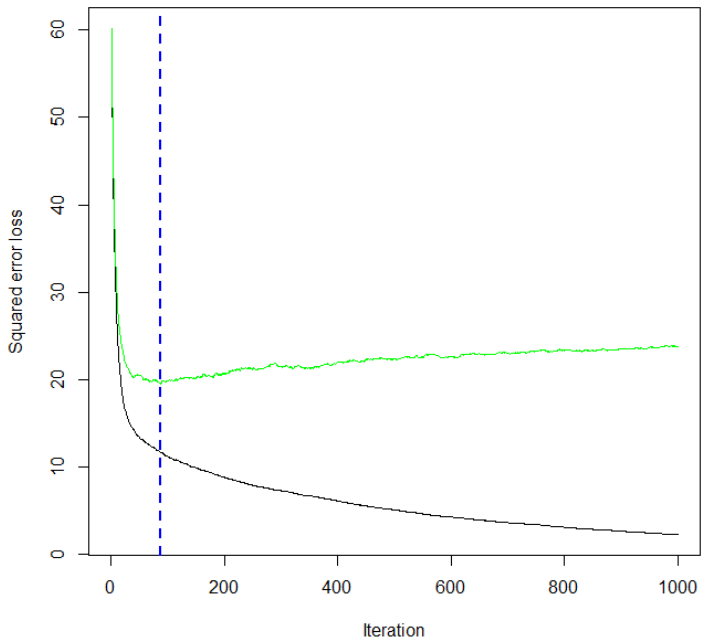
- formula - regression formulae
- data - dataset
- distribution - type of response variable
  ”gaussian” (squared error), ”laplace” (absolute loss), ”tdist” (t-distribution loss), ”bernoulli” (logistic regression for 0-1 outcomes), ”huberized” (huberized hinge loss for 0-1 outcomes), classes), ”adaboost” (the AdaBoost exponential loss for 0-1 outcomes), ”poisson” (count outcomes)
- n.tree - max number of trees J
- shrinkage - learning rate ν
- cv.folds - number of cross validation folds

Other specifications are found in CRAN.[9]

[9] https://cran.r-project.org/web/packages/gbm/gbm.pdf

Ozone data

With $\nu = 0.1$ and the max number of trees is 1000, the cv error against the number of trees J.



The optimal number of tree is 87 and the mse is 11.70987. This is an improvement compared to the optimal regression tree (mse or 15.18653).

Ozone data

Relative variable importance in the boosting tree

	var	rel.inf
x4.x7	x4.x7	17.4639361
x7	x7	17.2924256
x6.x7	x6.x7	13.6504837
x8.x10	x8.x10	7.8861462
x9.x9	x9.x9	5.4687928
x5.x7	x5.x7	3.3402348
x6.x10	x6.x10	2.9612786
x4	x4	2.7223048
x5.x6	x5.x6	2.6637839
x5.x8	x5.x8	2.4069812
x4.x10	x4.x10	2.2828979
x6.x8	x6.x8	2.1929048
x7.x8	x7.x8	2.1847759
x4.x6	x4.x6	1.9944113
x5.x10	x5.x10	1.7606405
x4.x5	x4.x5	1.6207382
x7.x10	x7.x10	1.5532738
x7.x9	x7.x9	1.5139316
x6	x6	1.4306353
x9.x10	x9.x10	1.3766736
x5.x9	x5.x9	1.2874346
x8.x9	x8.x9	1.0234030
x8	x8	0.9873060
x4.x8	x4.x8	0.8043989
x6.x9	x6.x9	0.6742396
x5	x5	0.4783302
x4.x9	x4.x9	0.4202396
x10	x10	0.3431101
x9	x9	0.2142874
x4.x4	x4.x4	0.0000000
x5.x5	x5.x5	0.0000000
x6.x6	x6.x6	0.0000000
x7.x7	x7.x7	0.0000000
x8.x8	x8.x8	0.0000000
x10.x10	x10.x10	0.0000000

Reference

- The Elements of Statistical Learning. Hastie, T., Tibshirani, R., Friedman, J. Spring Series in Statistics. `http://web.stanford.edu/~hastie/ElemStatLearn/`
- Extending the linear model with R. Faraway, J. J. (2006) CHAPMAN & HALL/CRC.
- Statistical Learning from a Regression Perspective. Berk, R. A. (2008) Springer.
- Boosting: Foundations and Algorithms. Schapire, R. E. and Freund, Y. (2012). MIT Press.
- https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf