Tree-Based Methods

生醫光電所 吳育德

A Concise Introduction to Machine Learning, 2020 Anita C. Faul, Chapter 5
Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning, Chapter 8

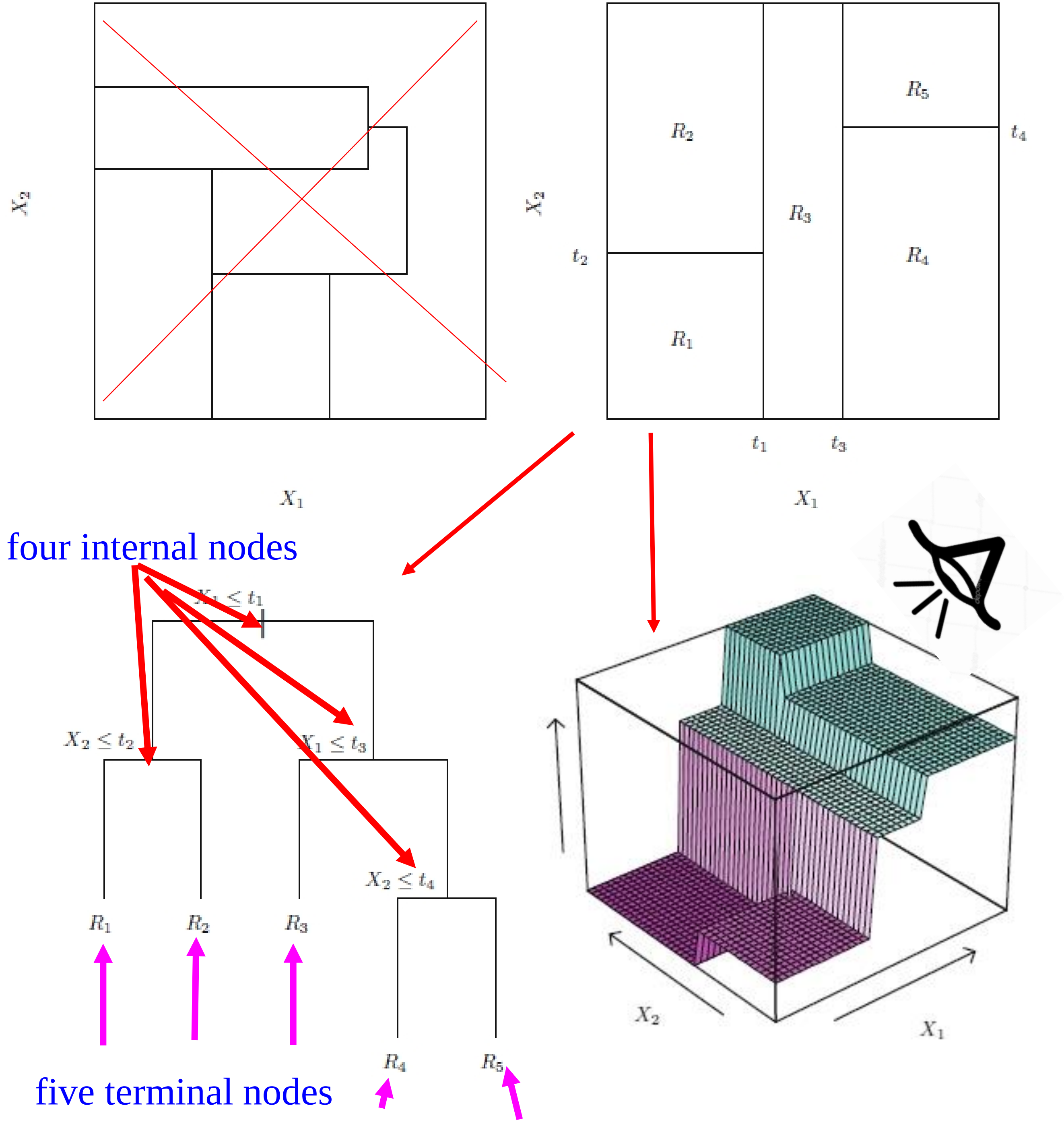
Decision Trees

- Decision Trees are binary trees consisting of
  - non-terminal nodes having two branches,
  - terminal nodes or leaves which are assigned a class.

- A sample enters the tree at the root node at the top.

- At each node, a decision is made whether the value of a particular feature is larger or smaller than a threshold.

- The sample traverses the tree down to a leaf and assigned that class.



Top Left: A partition of 2D feature space that could not result from recursive binary splitting.
Top Right: The output of recursive binary splitting on a 2D example.
Bottom Left: A tree corresponding to the partition in the top right panel.
Bottom Right: A perspective plot of the prediction surface corresponding to that tree.

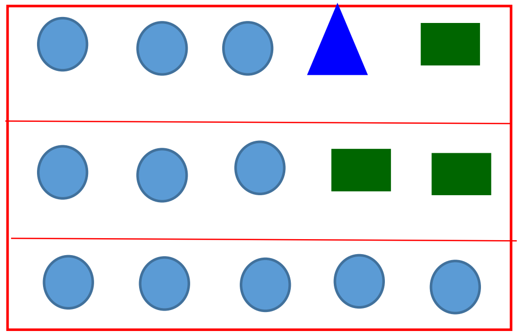
Classification Trees

- Predict that each observation belongs to ***the most commonly occurring class*** of training observations in the region.
- Use the recursive binary splitting to grow a classification tree.
- ***Gini index***

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

a measure of total variance (impurity) across K classes. $\hat{p}_{mk}$ represents the proportion of training observations in the mth region from the kth class

- Gini index is a measure of node ***purity***—a small value indicates that a node contains predominantly observations from a single class.



G$= \frac{3}{5}\frac{2}{5} + \frac{1}{5}\frac{4}{5} + \frac{1}{5}\frac{4}{5} = \frac{14}{25}$

G$= \frac{3}{5}\frac{2}{5} + \frac{2}{5}\frac{3}{5} = \frac{12}{25}$ less variant !

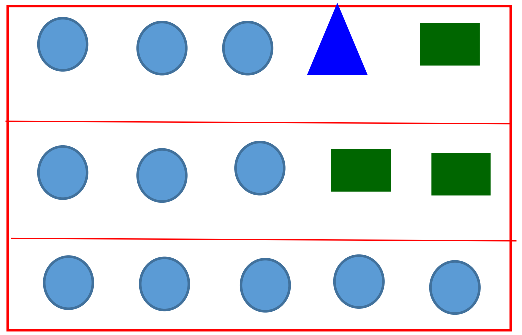
G$= \frac{5}{5}\frac{0}{5} = 0$ pure !

Entropy

- An alternative to the Gini index is ***entropy***, given by

$$H = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

- It turns out that the Gini index and the cross-entropy are quite similar numerically.



H$= -\frac{3}{5} log_2\left(\frac{3}{5}\right) - \frac{1}{5} log_2\left(\frac{1}{5}\right) - \frac{1}{5} log_2\left(\frac{1}{5}\right) = 1.371$

H$= -\frac{3}{5} log_2\left(\frac{3}{5}\right) - \frac{2}{5} log_2\left(\frac{2}{5}\right) = 0.971$ less variant !

H$= -\frac{5}{5} log_2\left(\frac{5}{5}\right) = 0$ pure !

Select the feature producing the highest Information gain

1. Compute entropy for a dataset with respect to a target feature

$$H(t, D) = -\sum_{l \in levels(t)} (P(t = l) \times log_2(P(t = l)))$$

where *levels*(*t*) is the set of levels of the target feature *t*, and *P*(*t* = *l*) is the probability of a randomly selected instance having the target feature level *l*.

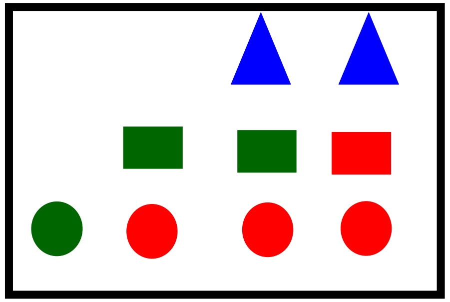
2. Use a particular feature *d* to create partitions $D_{d=l_1}, \cdots, D_{d=l_k}$, where $l_1, \cdots, l_k$ are the *k* levels that feature *d* can take. Each partition, $D_{d=l_i}$, contains the instances in that have a value of level l_i for the *d* feature. Compute the entropy remaining after partition

$$rem(t, D) = \sum_{l \in levels(t)} \underbrace{\frac{|D_{d=l}|}{|D|}}_{\text{weighting}} \times \underbrace{H(t, D_{d=l})}_{\substack{entropy\ of \\ Partition\ D_{d=l}}}$$

3. Information gain made from splitting the dataset using the feature *d*

$$IG(t, D) = H(t, D) - rem(t, D)$$

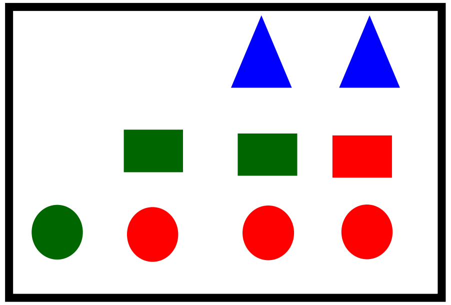
Choose a color feature to classify the shapes

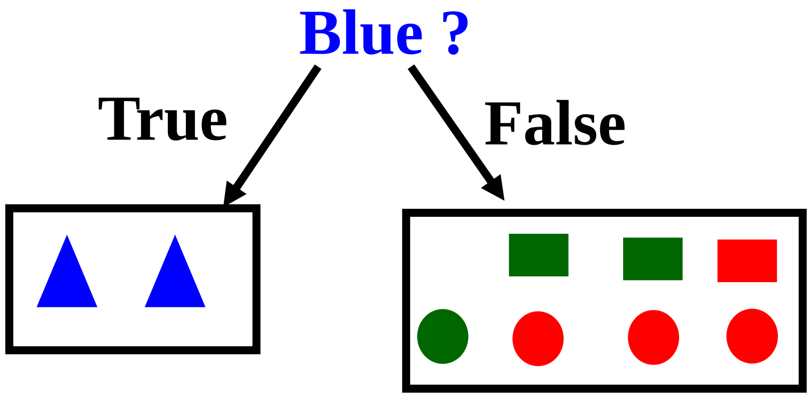


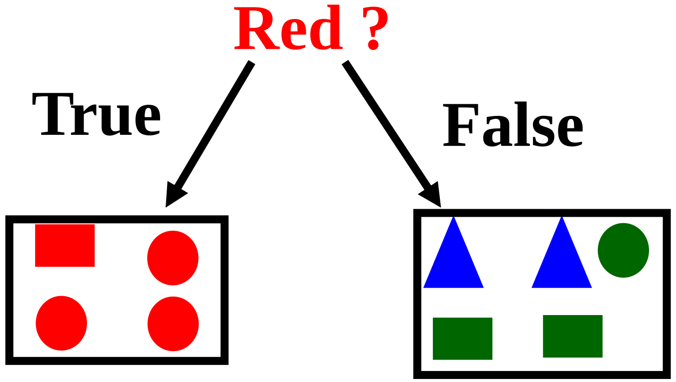
$H(shape: circle, square, triangle)$

$= -\frac{4}{9}log_2\left(\frac{4}{9}\right) - \frac{3}{9}log_2\left(\frac{3}{9}\right) - \frac{2}{9}log_2\left(\frac{2}{9}\right)$

$= 1.5305$







$H(triangle)$

$= -\frac{2}{2}log_2\left(\frac{2}{2}\right)$

$= 0$

Partition entropy

$H(circle, square)$

$= -\frac{4}{7}log_2\left(\frac{4}{7}\right) - \frac{3}{7}log_2\left(\frac{3}{7}\right)$

$= 0.9852$

Partition entropy

$H(circle, square)$

$= -\frac{3}{4}log_2\left(\frac{3}{4}\right) - \frac{1}{4}log_2\left(\frac{1}{4}\right)$

$= 0.8113$

Partition entropy

$H(circle, square, triangle)$

$= -\frac{1}{5}log_2\left(\frac{1}{5}\right) - \frac{2}{5}log_2\left(\frac{2}{5}\right)$

$-\frac{2}{5}log_2\left(\frac{2}{5}\right) = 1.5219$

Partition entropy

Entropy remaining: Rem $= \frac{2}{9} \times 0 + \frac{7}{9} \times 0.9852 = 0.7663$

Information gain: IG $= 1.5305 - \frac{7}{9} \times 0.9852 = 0.7642$

Rem $= \frac{4}{9} \times 0.8113 + \frac{5}{9} \times 1.5219 = 1.2061$

IG $= 1.5305 - 1.2061 = 0.3244$

Example 1

A convicted criminal who reoffends after release is known as a recidivist. The dataset describes prisoners released on parole, and whether they reoffended within two years of release.

ID	Good Behavior	Age < 30	Drug Dependent	Recidivist
1	false	true	false	true
2	false	false	false	false
3	false	true	false	true
4	true	false	false	false
5	true	false	true	true
6	true	false	false	false

Chap4. Exercise 2. John D. Kelleher, Brian Mac Namee, Aoife D'Arcy, **FUNDAMENTALS OF MACHINE LEARNING FOR PREDICTIVE DATA ANALYTICS, 2015**

- The first step: figure out which of the three features is the best one on which to split the dataset at the root node (i.e., which descriptive feature has the highest information gain).

- The total entropy for this dataset is

$$
\begin{aligned}
&H(\textsc{Recidivist}, \mathcal{D}) \\
&= - \sum_{l \in \left\{ \substack{true, \\ false} \right\}} P(\textsc{Recidivist} = l) \times log_2 \left(P(\textsc{Recidivist} = l) \right) \\
&= - \left(\left({}^3/_6 \times log_2({}^3/_6) \right) + \left({}^3/_6 \times log_2({}^3/_6) \right) \right) = 1.00\ bit
\end{aligned}
$$

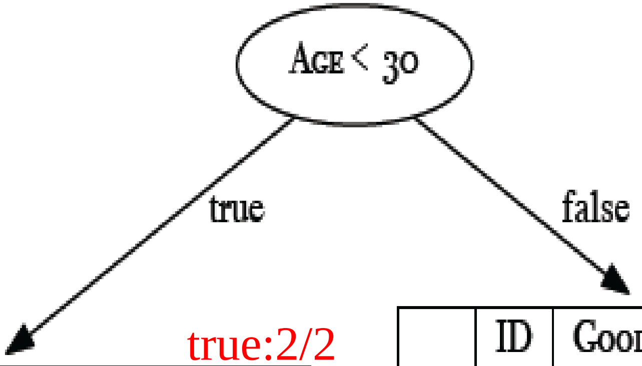
Chap4. Exercise 2. John D. Kelleher, Brian Mac Namee, Aoife D’Arcy, **FUNDAMENTALS OF MACHINE LEARNING FOR PREDICTIVE DATA ANALYTICS, 2015**

The table below illustrates the information gain for features:

Split by Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain	
GOOD BEHAVIOR	*true*	$\mathcal{D}_1$	$\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$	0.9183	0.9183	0.0817	=1.00-0.9183
	false	$\mathcal{D}_2$	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$	0.9183			
AGE $<$ 30	*true*	$\mathcal{D}_3$	$\mathbf{d}_1, \mathbf{d}_3$	0	0.5409 =0*2/6+.8113*4/6	$\boxed{0.4591}$	=1.00-0.5409
	false	$\mathcal{D}_4$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$	0.8113			
DRUG DEPENDENT	*true*	$\mathcal{D}_5$	$\mathbf{d}_5$	0	0.8091 =0*1/6+.9709*5/6	0.1909	=1.00-0.8091
	false	$\mathcal{D}_6$	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_6$	0.9709			

$$H(\text{Recidivist}) = -\frac{2}{2} log_2\left(\frac{2}{2}\right) = 0$$

Partition Entropy



$$H(\text{Recidivist}) = -\frac{1}{4} log_2\left(\frac{1}{4}\right) - \frac{3}{4} log_2\left(\frac{3}{4}\right) = 0.81130$$

Partition Entropy

true:2/2

	ID	GOOD BEHAVIOR	DRUG DEPENDENT	RECIDIVIST
D3	1	false	false	true
	3	false	false	true

No further split

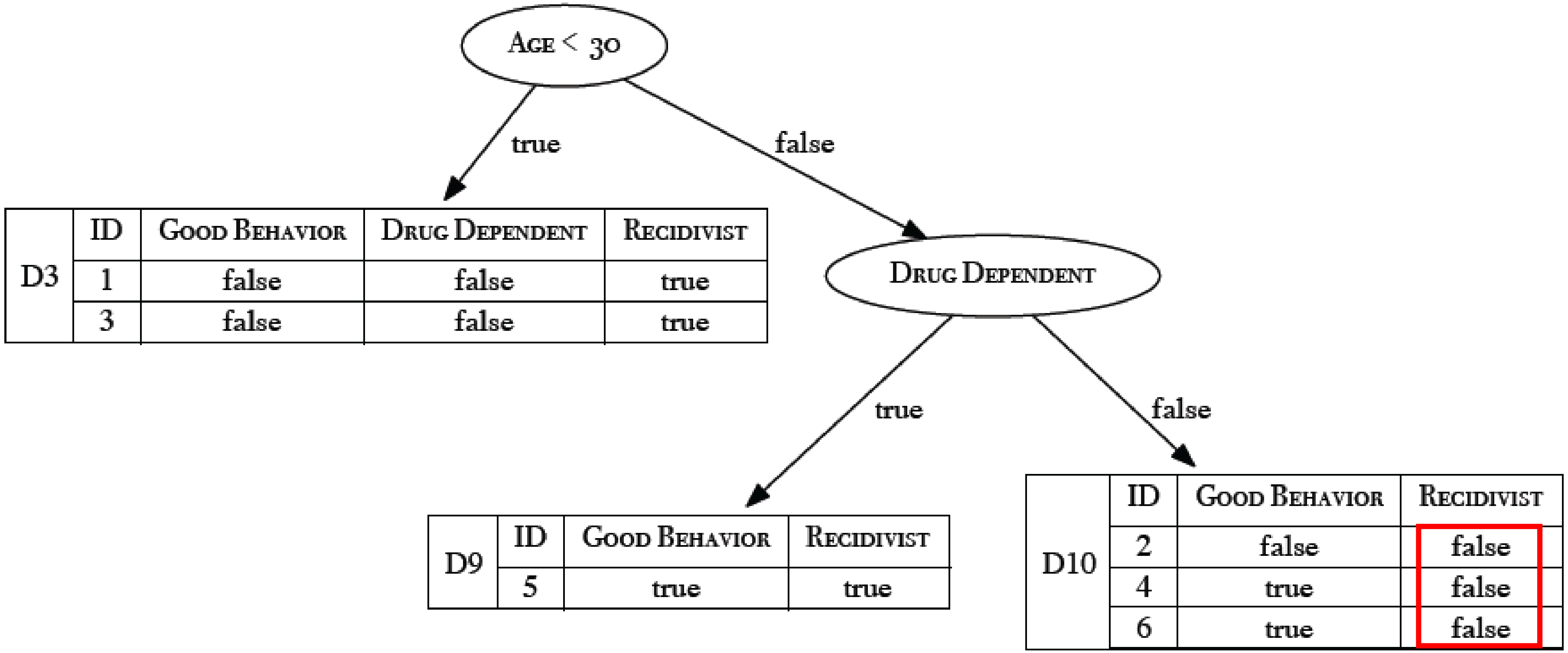
	ID	GOOD BEHAVIOR	DRUG DEPENDENT	RECIDIVIST
D4	2	false	false	false
	4	true	false	false
	5	true	true	true
	6	true	false	false

true:1/4
False:3/4

- The dataset on the right branch of the tree (D_4) is not homogenous, so we need to grow this branch of the tree. The entropy for this dataset, D_4, is:

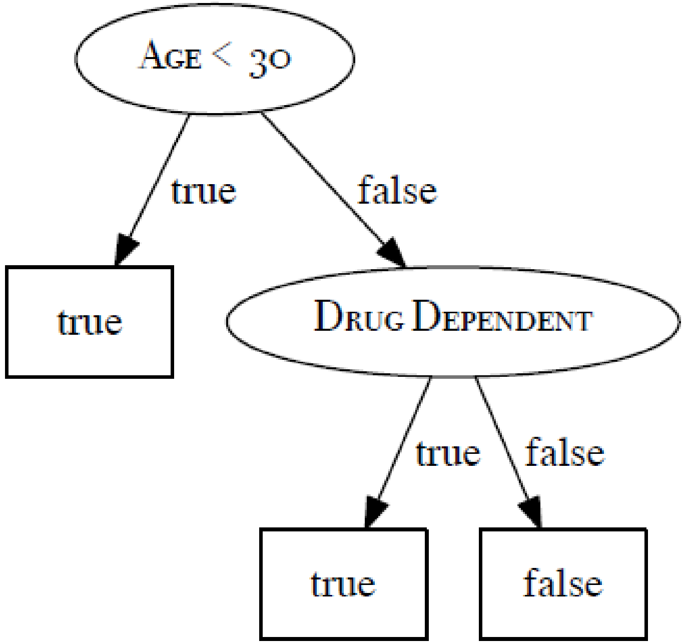
$$H(\text{RECIDIVIST}, \mathcal{D}_4)$$
$$= -\sum_{l \in \left\{\begin{matrix} true, \\ false \end{matrix}\right\}} P(\text{RECIDIVIST} = l) \times log_2(P(\text{RECIDIVIST} = l))$$
$$= -\left(\left({}^1/_4 \times log_2({}^1/_4)\right) + \left({}^3/_4 \times log_2({}^3/_4)\right)\right) = 0.8113 \; bits$$

Split by Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain	
GOOD BEHAVIOR	*true*	$\mathcal{D}_7$	$\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$	0.918295834	0.4591	0.3522	=0.8113-0.4591
	false	$\mathcal{D}_8$	$\mathbf{d}_2$	0			
DRUG DEPENDENT	*true*	$\mathcal{D}_9$	$\mathbf{d}_5$	0	0	$\boxed{0.8113}$	=0.8113-0
	false	$\mathcal{D}_{10}$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_6$	0			



No further split

AGE <30 = false,
DRUG DEPENDENT = true
GOOD BEHAVIOR = false,
→RECIDIVIST = true



AGE <30 = true,
GOOD BEHAVIOR = true,
DRUG DEPENDENT = false
→RECIDIVIST = true

Example 2

ID	Age	Education	Marital Status	Occupation	Annual Income
1	39	bachelors	never married	transport	$25K$–$50K$
2	50	bachelors	married	professional	$25K$–$50K$
3	18	high school	never married	agriculture	$<25K$
4	28	bachelors	married	professional	$25K$–$50K$
5	37	high school	married	agriculture	$25K$–$50K$
6	24	high school	never married	armed forces	$<25K$
7	52	high school	divorced	transport	$25K$–$50K$
8	40	doctorate	married	professional	$>50K$

OCCUPATION :
transport = works in the transportation industry;
professional= doctors, lawyers, etc.;
agriculture = works in the agricultural industry;
armed forces = is a member of the armed forces;

Chap4. Exercise 3. John D. Kelleher, Brian Mac Namee, Aoife D’Arcy, **FUNDAMENTALS OF MACHINE LEARNING FOR PREDICTIVE DATA ANALYTICS, 2015**

Calculate the entropy:

$$
\begin{aligned}
&H(\textsc{Annual Income}, \mathcal{D}) \\
&= - \sum_{l \in \left\{ \substack{<25K, \\ 25K\text{–}50K, \\ >50K} \right\}} P(\textsc{An. Inc.} = l) \times log_2 \left(P(\textsc{An. Inc.} = l) \right) \\
&= - \left(\left(\frac{2}{8} \times log_2 \left(\frac{2}{8} \right) \right) + \left(\frac{5}{8} \times log_2 \left(\frac{5}{8} \right) \right) + \left(\frac{1}{8} \times log_2 \left(\frac{1}{8} \right) \right) \right) \\
&= 1.2988 \; bits
\end{aligned}
$$

Calculate the Gini index:

$$
\begin{aligned}
&Gini(\textsc{Annual Income}, \mathcal{D}) \\
&= 1 - \sum_{l \in \left\{ \substack{<25K, \\ 25K\text{–}50K, \\ >50K} \right\}} P(\textsc{An. Inc.} = l)^2 \\
&= 1 - \left(\left(\frac{2}{8} \right)^2 + \left(\frac{5}{8} \right)^2 + \left(\frac{1}{8} \right)^2 \right) = 0.5313
\end{aligned}
$$

First sort the instances according to the AGE feature:

ID	AGE	ANNUAL INCOME
3	18	<25K
6	24	<25K
4	28	25K–50K
5	37	25K–50K
1	39	25K–50K
8	40	>50K
2	50	25K–50K
7	52	25K–50K

26

39.5

45

The mid-points in the AGE values that are adjacent in the new ordering but that have different target levels define the possible threshold points: 26, 39.5, and 45.

Split by Feature	Partition	Instances	Partition Entropy	Rem.	Info. Gain
>26	$\mathcal{D}_1$	$\mathbf{d}_3, \mathbf{d}_6$	0	0.4875	$\boxed{0.8113}$
	$\mathcal{D}_2$	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_7, \mathbf{d}_8$	0.6500		
>39.5	$\mathcal{D}_3$	$\mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$	0.9710	0.9456	0.3532
	$\mathcal{D}_4$	$\mathbf{d}_2, \mathbf{d}_7, \mathbf{d}_8$	0.9033		
>45	$\mathcal{D}_5$	$\mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6, \mathbf{d}_8$	1.4591	1.0944	0.2044
	$\mathcal{D}_6$	$\mathbf{d}_2, \mathbf{d}_7$	0		

Split by Feature	Level	Instances	Partition Entropy	Rem.	Info. Gain
EDUCATION	*high school*	$\mathbf{d}_3, \mathbf{d}_5, \mathbf{d}_6, \mathbf{d}_7$	1.0	0.5	0.7988
	bachelors	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$	0		
	doctorate	$\mathbf{d}_8$	0		
MARITAL STATUS	*never married*	$\mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_6$	0.9183	0.75	0.5488
	married	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_8$	0.8113		
	divorced	$\mathbf{d}_7$	0		
OCCUPATION	*transport*	$\mathbf{d}_1, \mathbf{d}_7$	0	0.5944	0.7044
	professional	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_8$	0.9183		
	agriculture	$\mathbf{d}_3, \mathbf{d}_5$	1.0		
	armed forces	$\mathbf{d}_6$	0		

Chap4. Exercise 3. John D. Kelleher, Brian Mac Namee, Aoife D’Arcy, **FUNDAMENTALS OF MACHINE LEARNING FOR PREDICTIVE DATA ANALYTICS, 2015**

Homework 1

Finish the tree splitting of Example 2.

Homework 2

Consider the following n = 16 points in two dimensions, training a binary tree using the entropy impurity.

1. Plot the points of $\boldsymbol{\omega_1}$ and points of $\boldsymbol{\omega_2}$ in the 2D x_1-x_2 plane.
2. Provide the step-by-step split feature Table similar to Example 1.
3. Illustrate the recursive binary splitting on the 2D x_1-x_2 plane.

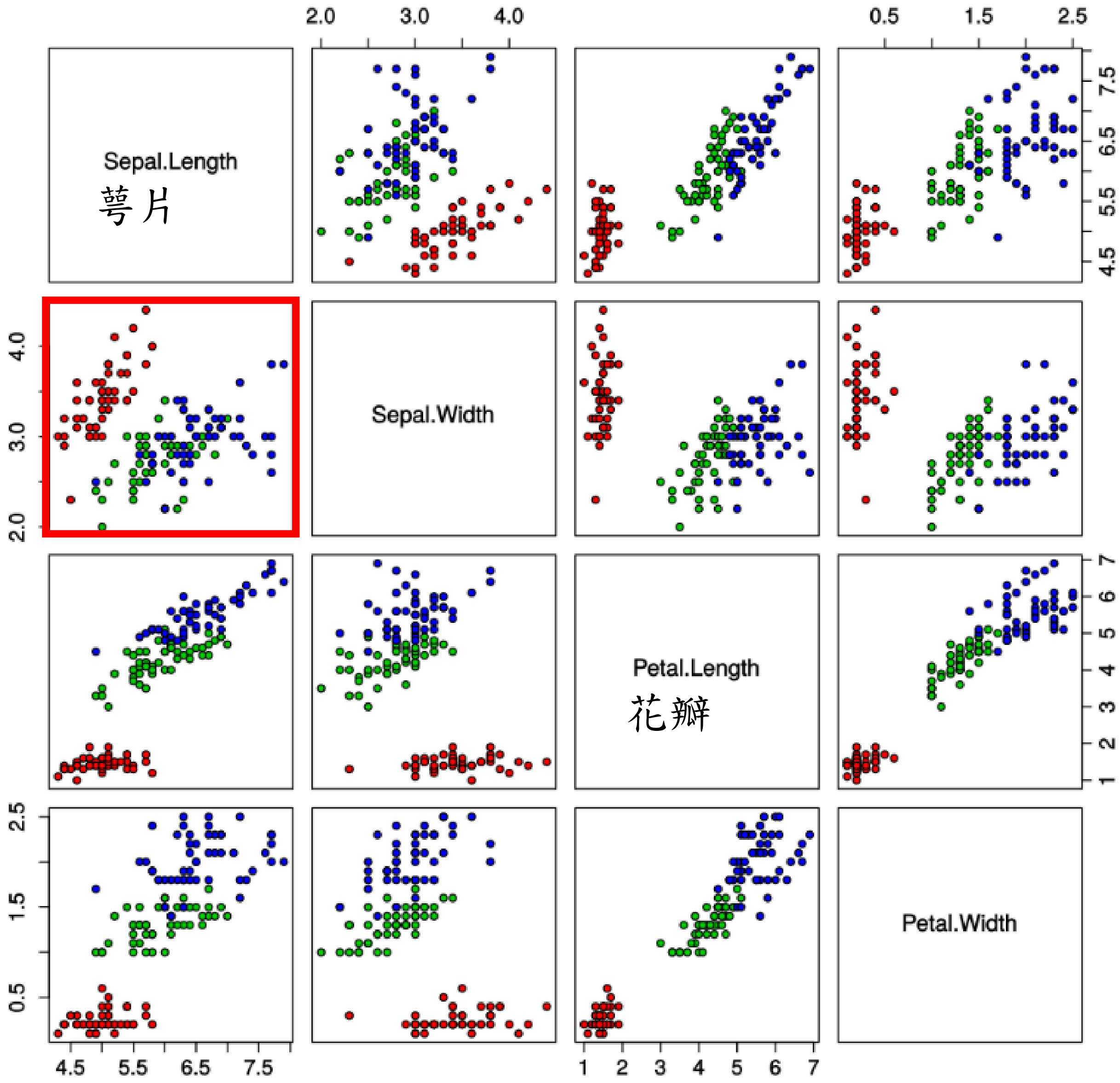
$\boldsymbol{\omega_1}$ **(black)**		$\boldsymbol{\omega_2}$ **(red)**	
x_1	x_2	x_1	x_2
.15	.83	.10	.29
.09	.55	.08	.15
.29	.35	.23	.16
.38	.70	.70	.19
.52	.48	.62	.47
.57	.73	.91	.27
.73	.75	.65	.90
.47	.06	.75	.36

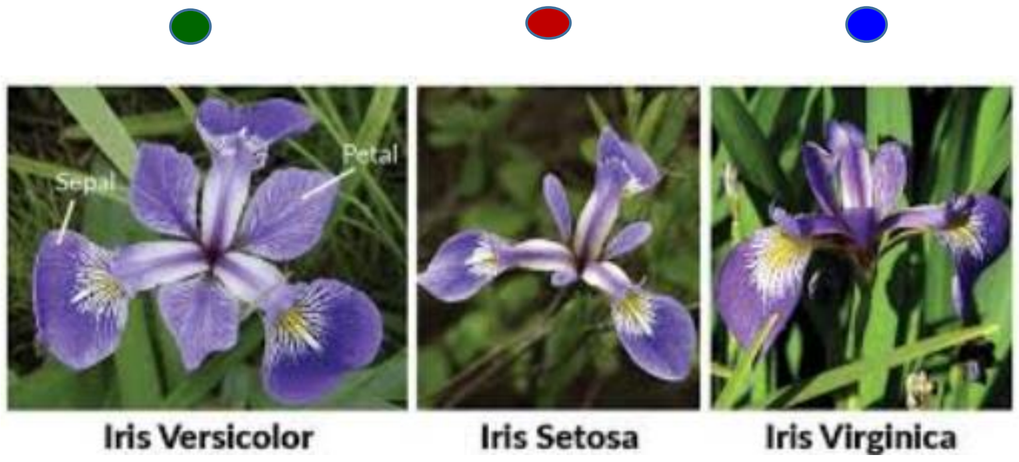
安德森鳶尾花卉數據集

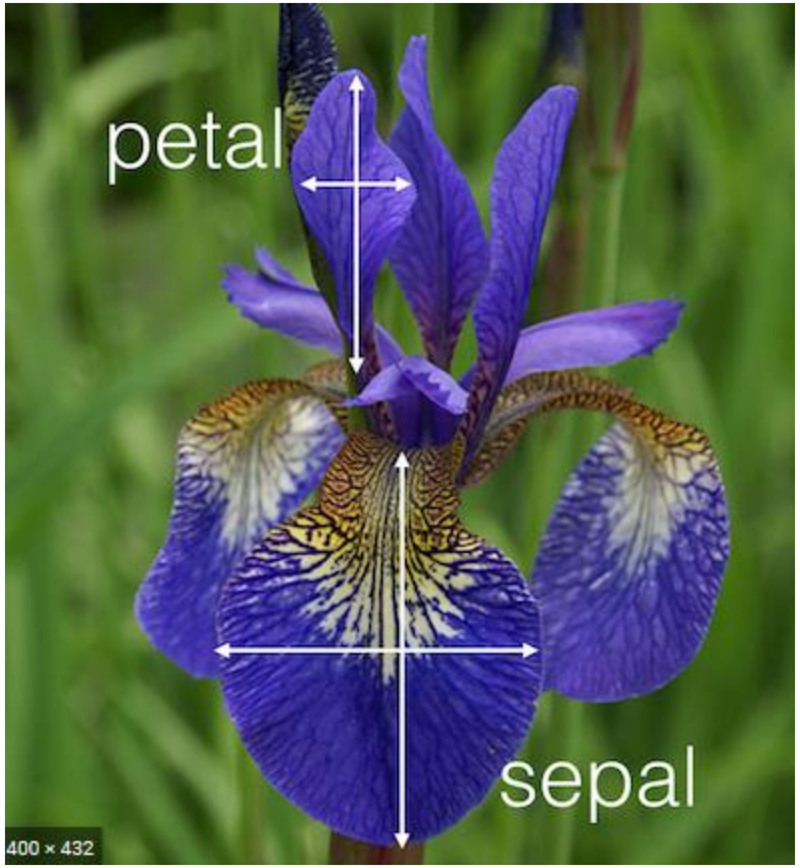
- **安德森鳶尾花卉數據集**（Anderson's Iris data set），也稱**鳶尾花卉數據集**（Iris flower data set）或**費雪鳶尾花卉數據集**（Fisher's Iris data set），是一類多重變量分析的數據集。
- 它最初是埃德加·安德森（Edgar Anderson）從加拿大加斯帕半島上的鳶尾屬花朵中提取的形態學變異數據，後由羅納德·費雪作為判別分析的一個例子，運用到統計學中。
- 其數據集包含了150個樣本，都屬於鳶尾屬下的三個亞屬，分別是山鳶尾、變色鳶尾和維吉尼亞鳶尾（Virginia Iris）。
- 四個特徵被用作樣本的定量分析，它們分別是花萼和花瓣的長度和寬度。

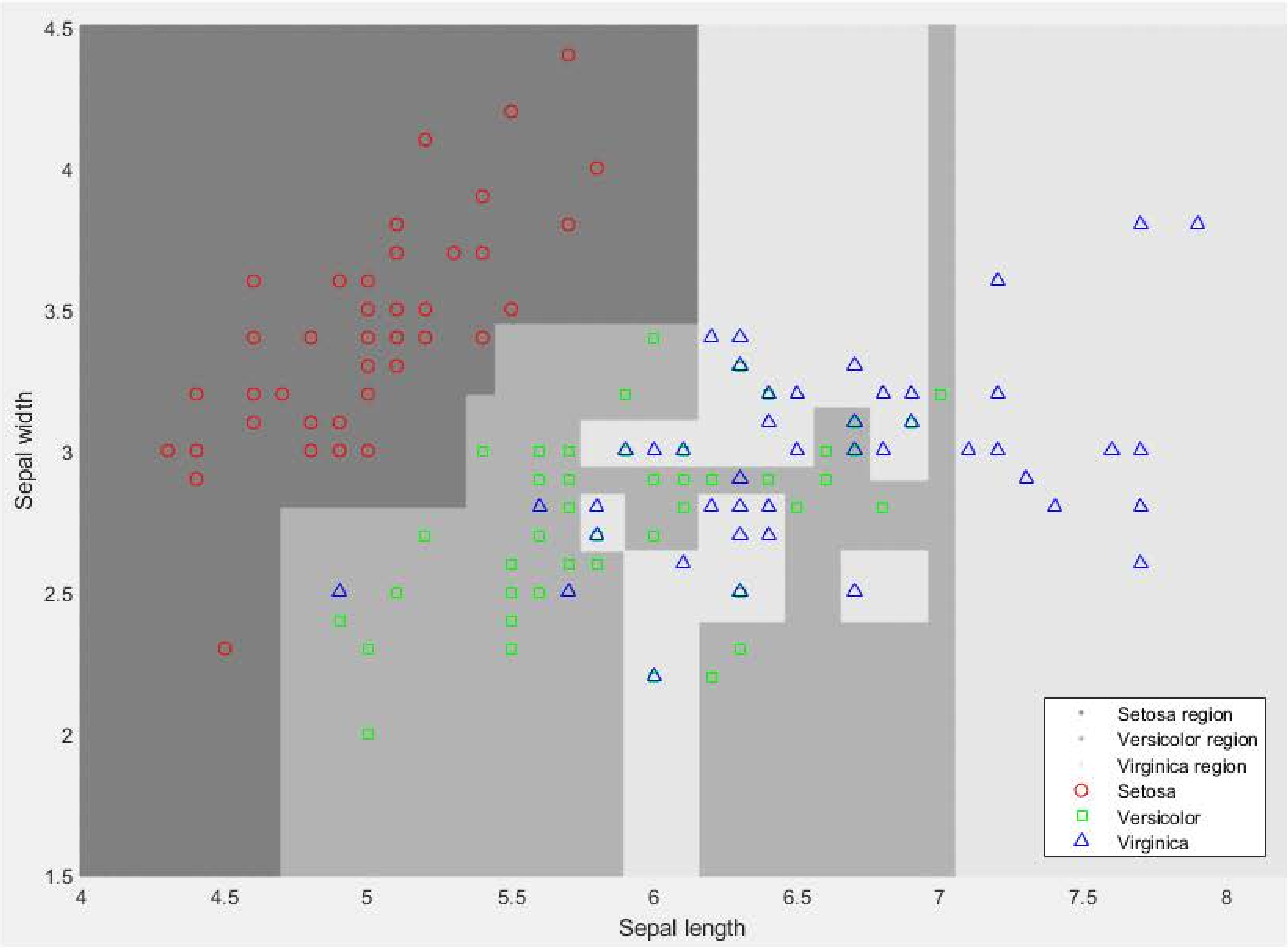
https://zh.wikipedia.org/wiki/%E5%AE%89%E5%BE%B7%E6%A3%AE%E9%B8%A2%E5%B0%BE%E8%8A%B1%E5%8D%89%E6%95%B0%E6%8D%AE%E9%9B%86

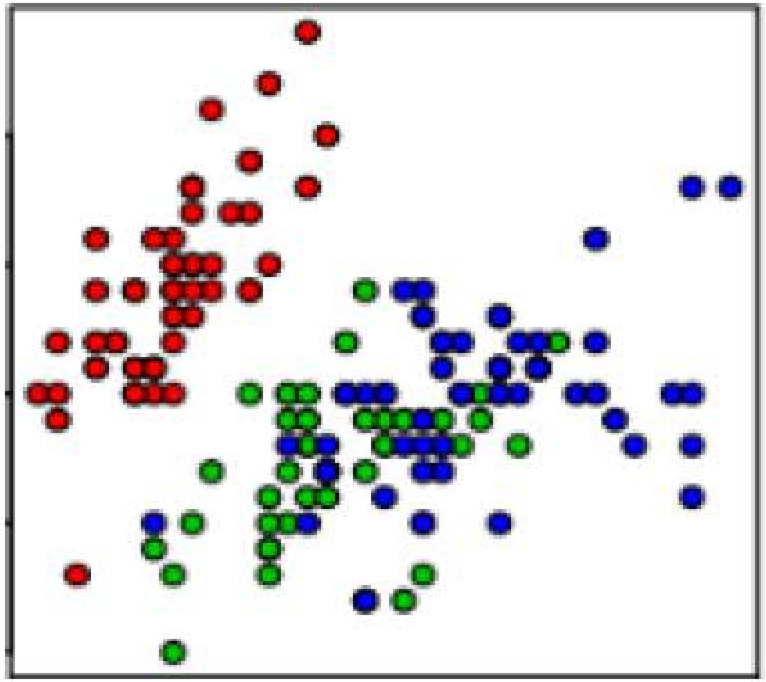
Iris Data (red=setosa,green=versicolor,blue=virginica)

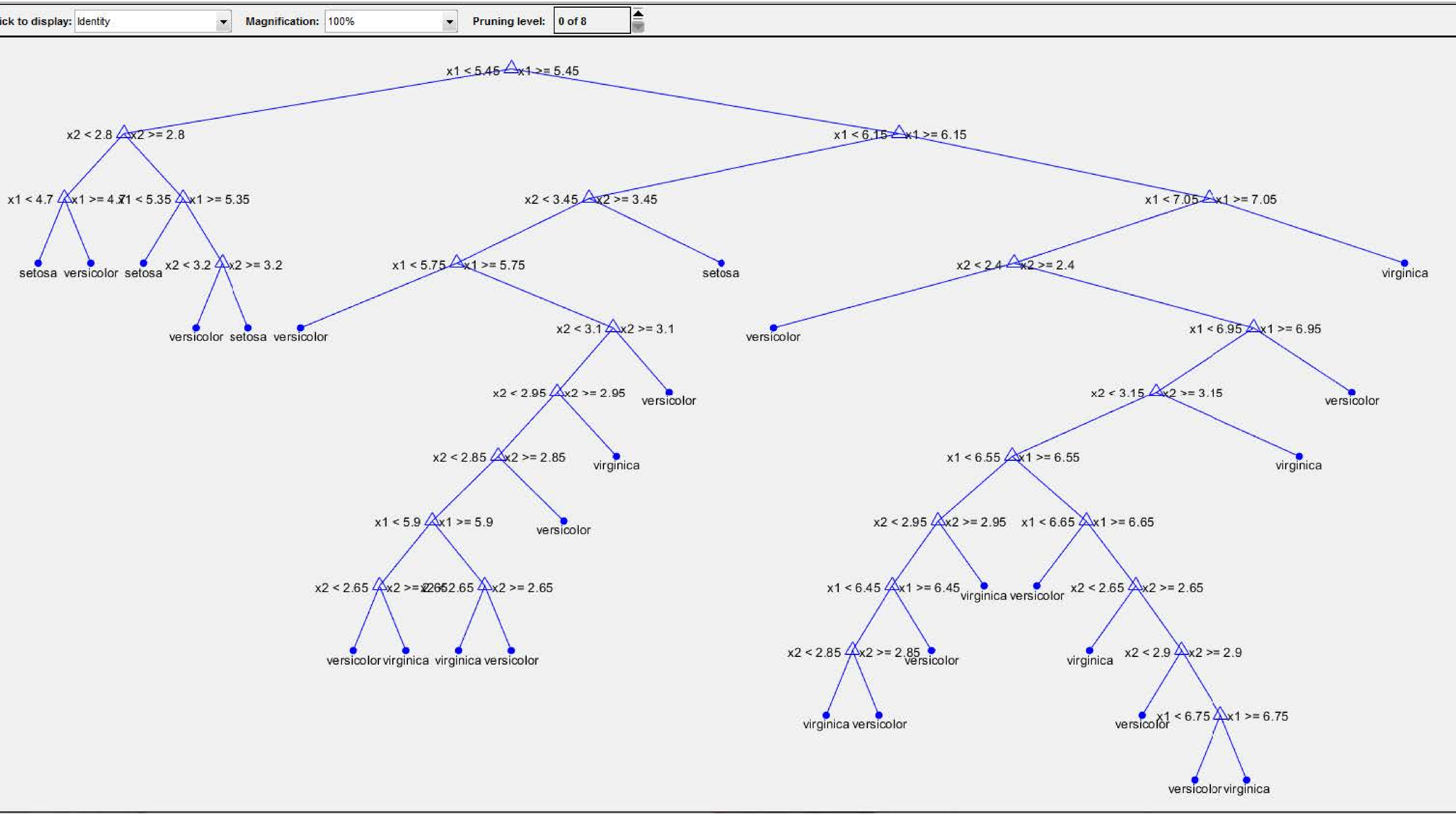







Sepal width
Sepal length
4.5
4
3.5
3
2.5
2
1.5
4
4.5
5
5.5
6
6.5
7
7.5
8
Setosa region
Versicolor region
Virginica region
Setosa
Versicolor
Virginica



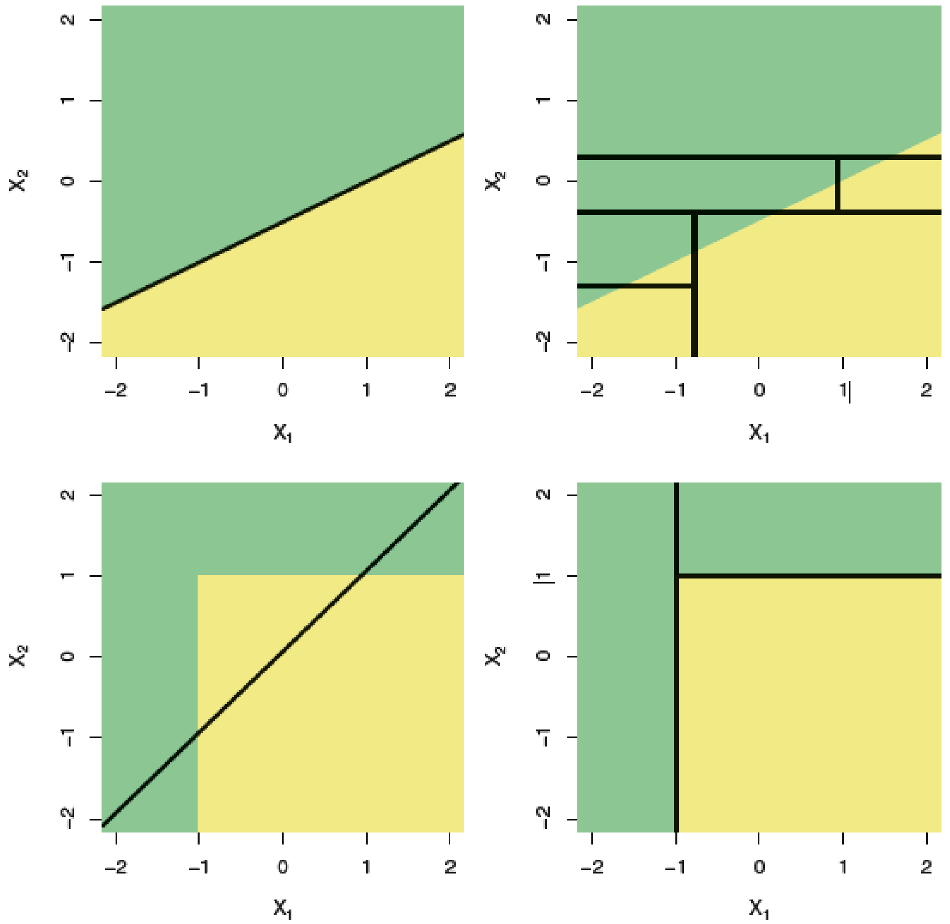


ick to display: Identity
Magnification: 100%
Pruning level: 0 of 8
x1 < 5.45 x1 >= 5.45
x2 < 2.8 x2 >= 2.8
x1 < 6.15 x1 >= 6.15
x1 < 4.7 x1 >= 4.7
x1 < 5.35 x1 >= 5.35
x2 < 3.45 x2 >= 3.45
x1 < 7.05 x1 >= 7.05
setosa
versicolor
setosa
x2 < 3.2 x2 >= 3.2
x1 < 5.75 x1 >= 5.75
setosa
x2 < 2.4 x2 >= 2.4
virginica
versicolor
setosa
versicolor
x2 < 3.1 x2 >= 3.1
versicolor
x1 < 6.95 x1 >= 6.95
x2 < 2.95 x2 >= 2.95
versicolor
x2 < 3.15 x2 >= 3.15
versicolor
x2 < 2.85 x2 >= 2.85
virginica
x1 < 6.55 x1 >= 6.55
virginica
x1 < 5.9 x1 >= 5.9
versicolor
x2 < 2.95 x2 >= 2.95
x1 < 6.65 x1 >= 6.65
x2 < 2.65 x2 >= 2.65
x2 < 2.65 x2 >= 2.65
x1 < 6.45 x1 >= 6.45
virginica
versicolor
x2 < 2.65 x2 >= 2.65
versicolor
virginica
virginica
versicolor
x2 < 2.85 x2 >= 2.85
versicolor
virginica
x2 < 2.9 x2 >= 2.9
virginica
versicolor
versicolor
x1 < 6.75 x1 >= 6.75
versicolor
virginica

```matlab
% Fishertree.m from  A Concise Introduction to Machine Learning, 2020 Anita C. Faul
load fisheriris
% Extract two attributes.
sl = meas(:,1); % sepal length
sw = meas(:,2); % sepal width
X = [sl,sw];

% Create classifier.
% The depth of a decision tree is governed by three arguments:
% Maximum number of branch node splits; a large value results in a deep tree.
MaxNumSplits = size(X,1) - 1;

% Minimum number of samples per branch node; a small number results in a deep tree.
MinParentSize = 5;

% Minimum number of samples per leaf; a small number results in a deep tree.
MinLeafSize = 1;

treeModel = fitctree(X,species,...
    'MaxNumSplits',MaxNumSplits,...
    'MinLeafSize',MinLeafSize,...
    'MinParentSize',MinParentSize);
view(treeModel,'mode','graph') % visualization
```

```
% Lay grid over the region
d = 0.01;
[x1Grid,x2Grid] = meshgrid(4:d:8.2,1.5:d:4.5);
xGrid = [x1Grid(:),x2Grid(:)];  N = size(xGrid,1);
% For each grid point calculate the score of each class.
% 'predict' returns the predicted class labels corresponding to the
% minimum misclassification cost, the score (posterior probability)
% for each class as well as the predicted node number and class number.

[~,score,~,~] = predict(treeModel,xGrid);
% Classify according to the maximum score.
[~,maxScore] = max(score,[],2);

% Plot classifier regions.
figure
h(1:3) = gscatter(xGrid(:,1),xGrid(:,2),maxScore,...
    [0.5 0.5 0.5; 0.7 0.7 0.7; 0.9 0.9 0.9]);
hold on
% Plot data.
h(4:6) = gscatter(sl, sw, species,'rgb','os^');
xlabel('Sepal length'); ylabel('Sepal width');
legend(h,{'Setosa region','Versicolor region','Virginica region',...
          'Setosa','Versicolor','Virginica'},'Location','Southeast');
axis([4 8.2 1.5 4.5])
```

Tree Versus Linear Models



Top Row: A 2D classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).
Bottom Row: Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).

Advantages and Disadvantages of Trees

▲ Easy to explain to people !

▲ More closely mirror human decision making.

▲ Can be displayed graphically and easily interpreted even by a non-expert.

▲ Can easily handle qualitative predictors without creating dummy variables.

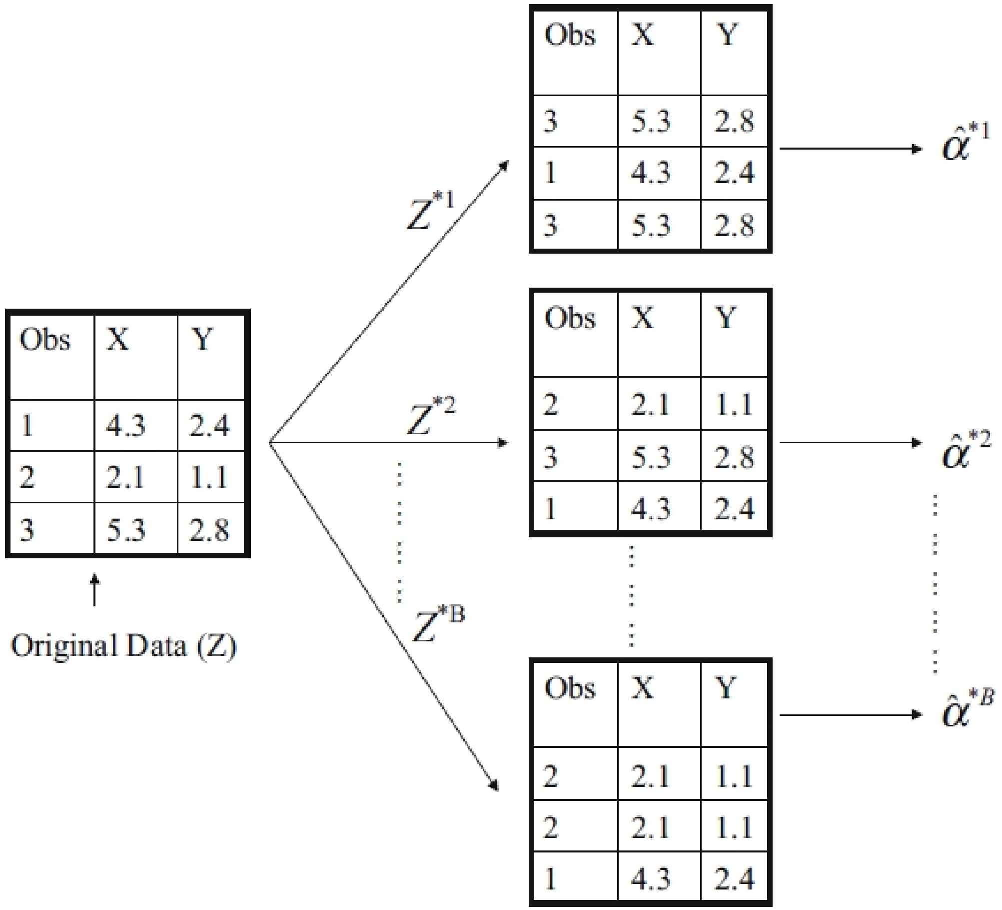
▼ Generally do not have the same level of predictive accuracy as some of the other classification approaches.

By aggregating many decision trees, the predictive performance of trees can be substantially improved.

Bagging

- The bootstrap is an extremely powerful idea. It is used in many situations in which it is hard or even impossible to directly compute the standard deviation of a quantity of interest.
- ***Bootstrap aggregation***, or ***bagging***, is a general-purpose procedure for reducing the variance of a statistical learning method.

Example with just 3 observations

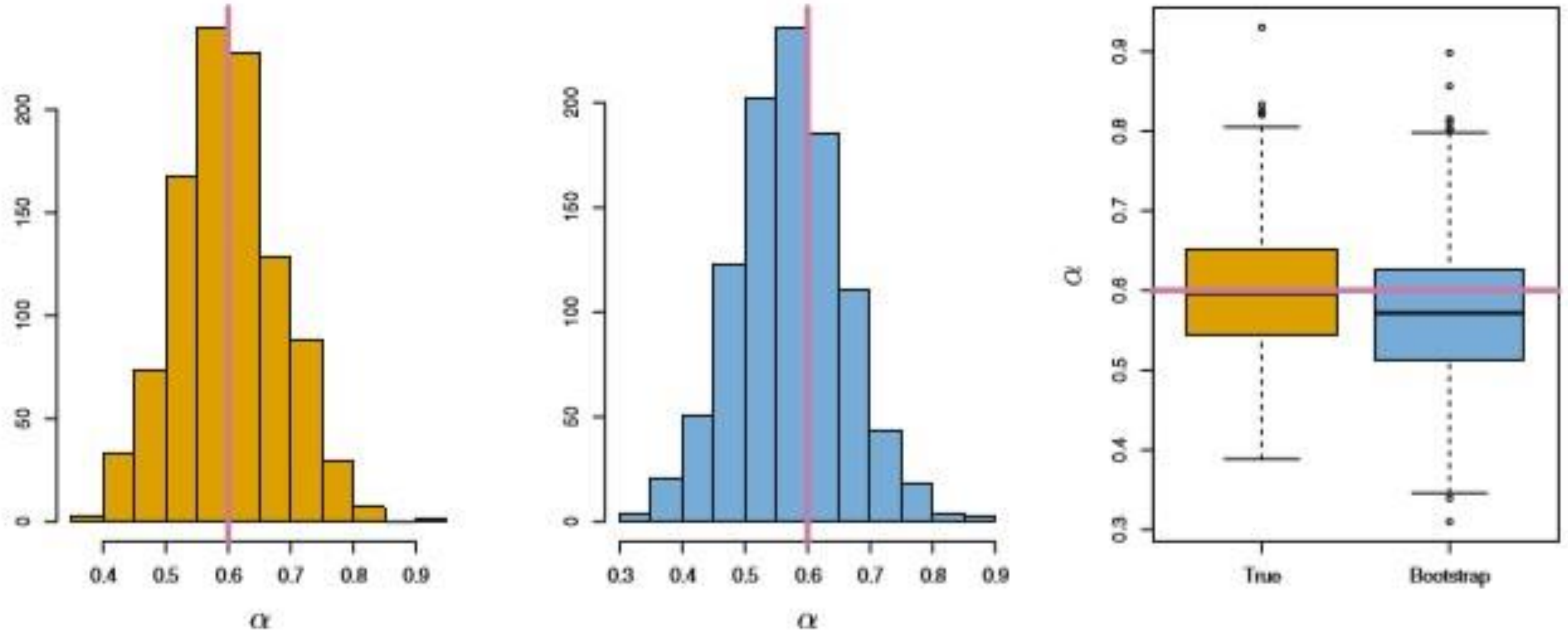


$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}.$$

A graphical illustration of the bootstrap approach on a small sample containing

n =3 observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α

Results



Left: A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population. Center: A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set. Right: The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.

Bagging classification trees

- Bootstrap by taking repeated samples from the training data set.
- First generate b different bootstrapped training data sets.
- Then train the jth bootstrapped training set to get the predictions $\boldsymbol{x}$ at $\varphi_j(\boldsymbol{x})$.
- We then average all the predictions to obtain

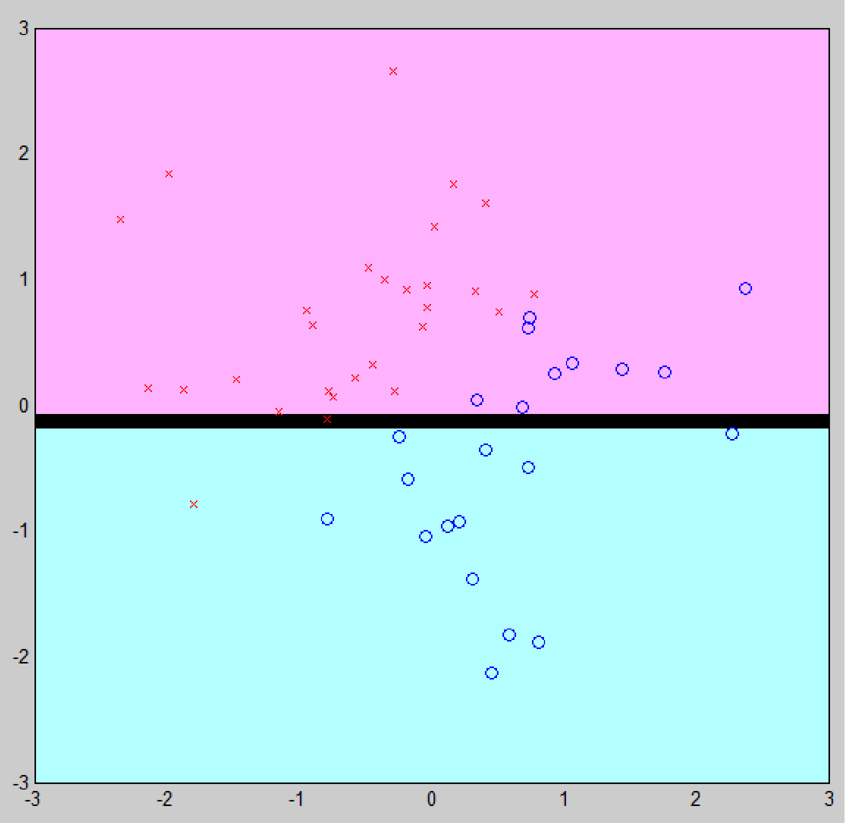
$$f \leftarrow \frac{1}{b}\sum_{j=1}^{b} \varphi_j(\boldsymbol{x})$$

This is called ***bagging***.

- For each test observation, we record the class predicted by each of the j trees, and take a ***majority vote***: the overall prediction is the most commonly occurring class among the B predictions.

MATLAB code for decision stump classification

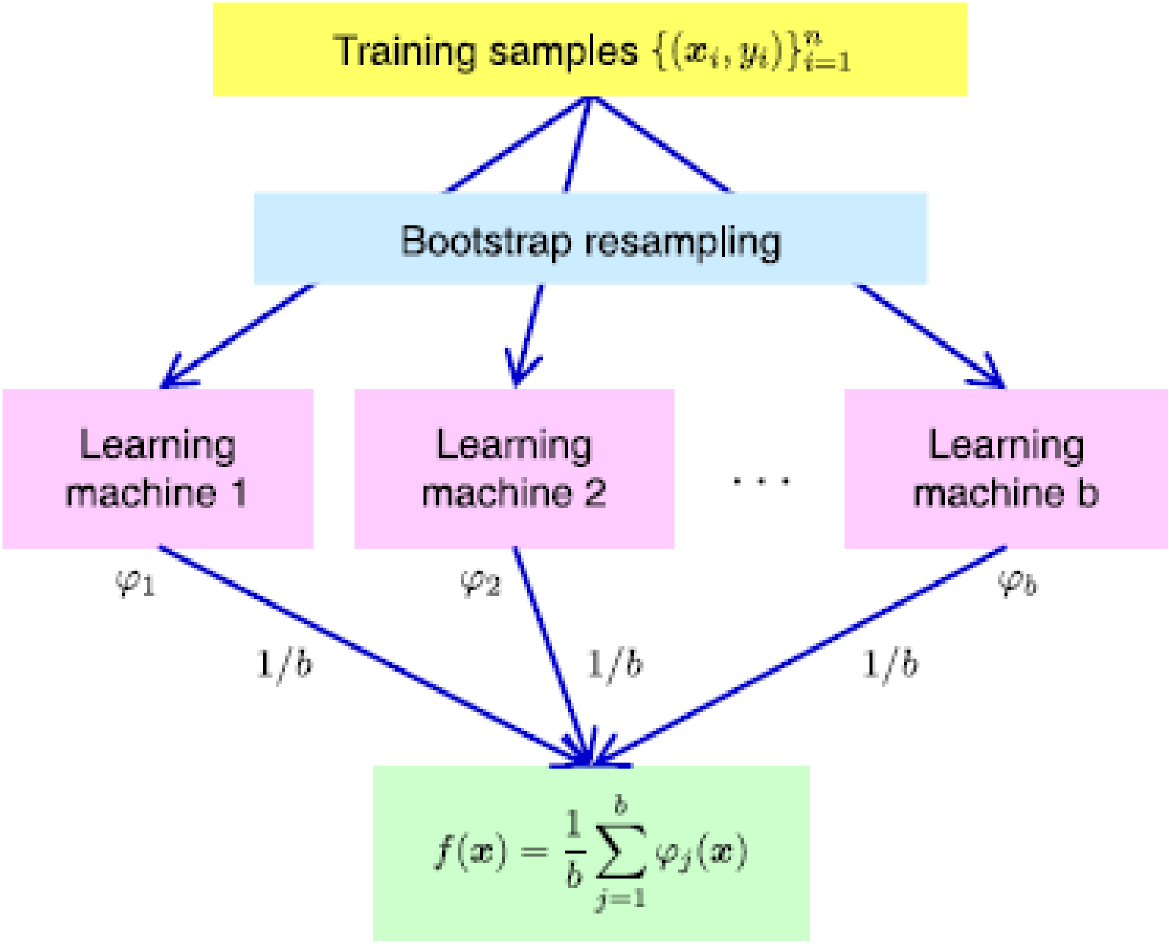
```
% A decision stump is a depth-one version of a decision tree.
% tree_stump.m
x=randn(50,2);
y=2*(x(:,1)>x(:,2))-1;
X0=linspace(-3,3,50);
[X(:,:,1) X(:,:,2)]=meshgrid(X0);
d=ceil(2*rand);
[xs,xi]=sort(x(:,d));
el=cumsum(y(xi));
eu=cumsum(y(xi(end:-1:1)));
e=eu(end-1:-1:1)-el(1:end-1);
[em,ei]=max(abs(e));
c=mean(xs(ei:ei+1));
s=sign(e(ei));
Y=sign(s*(X(:,:,d)-c));
figure(1); clf; hold on; axis([-3 3 -3 3]);
colormap([1 0.7 1; 0.7 1 1]); contourf(X0,X0,Y);
plot(x(y==1,1),x(y==1,2),'bo');
plot(x(y==-1,1),x(y==-1,2),'rx');
```



Bagging for decision stumps

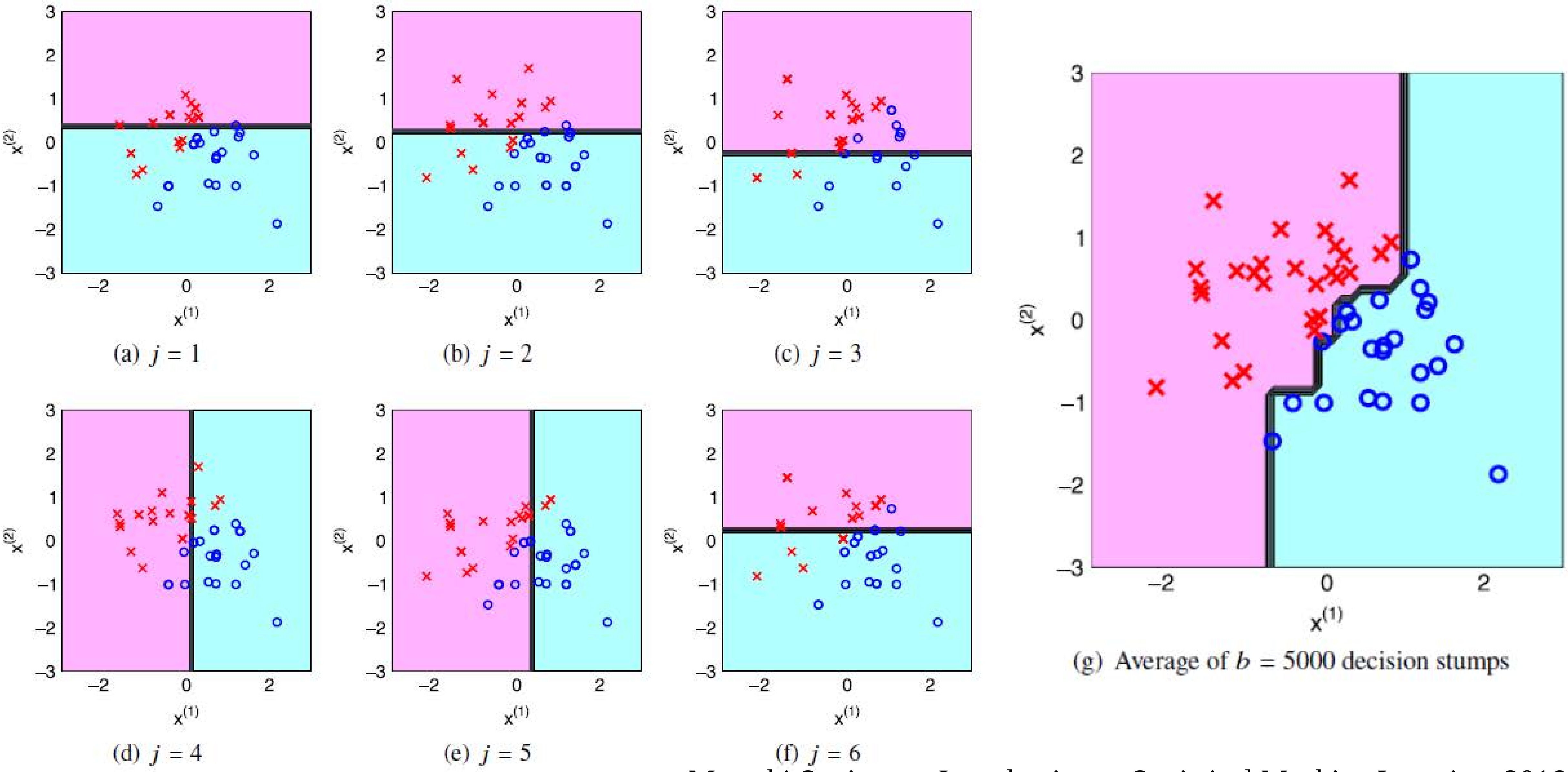
1. For $j = 1, \cdots, b$
(a) Randomly choose n samples from $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ with replacement.
(b) Train a classifier φ_j with the randomly resampled data set.
2. Output the average of $\{\varphi_j\}_{j=1}^{b}$ as the final solution f :

$$f \leftarrow \frac{1}{b} \sum_{j=1}^{b} \varphi_j(\boldsymbol{x})$$



Masashi Sugiyama, Introduction to Statistical Machine Learning, 2016

Example of bagging for decision stumps



Masashi Sugiyama, Introduction to Statistical Machine Learning, 2016

```matlab
% bagging for decision stumps
% bagging.m
n=50; x=randn(n,2);
y=2*(x(:,1)>x(:,2))-1;
b=5000; a=50; Y=zeros(a,a);
X0=linspace(-3,3,a);
[X(:,:,1) X(:,:,2)]=meshgrid(X0);
for j=1:b
  db=ceil(2*rand);
  r=ceil(n*rand(n,1));
  xb=x(r,:); yb=y(r);
  [xs,xi]=sort(xb(:,db));
  el=cumsum(yb(xi));
  eu=cumsum(yb(xi(end:-1:1)));
  e=eu(end-1:-1:1)-el(1:end-1);
  [em,ei]=max(abs(e)); c=mean(xs(ei:ei+1));
  s=sign(e(ei));
  Y=Y+sign(s*(X(:,:,db)-c))/b;
end
figure(1); clf; hold on; axis([-3 3 -3 3]);
colormap([1 0.7 1; 0.7 1 1]); contourf(X0,X0,sign(Y));
plot(x(y==1,1),x(y==1,2),'bo');
plot(x(y==-1,1),x(y==-1,2),'rx');
```

Random Forests

- ***Random forests*** provide an improvement over bagged trees by way of a random small tweak that ***decorrelates*** the trees. This reduces the variance when we average the trees.
- As is bagging, we build a number of decision trees on bootstrapped training samples.
- Each time a split in a tree, ***a random selection of m predictors*** is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors.
- A fresh selection of m predictors is taken at each split, and typically we choose $m \approx \sqrt{p}$—that is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors. For regression purpose, use $m \approx \frac{p}{3}$.

Out-of-Bag Error Estimation

- There is a very straightforward way to estimate the test error of a bagged model.
- The key to bagging is that trees are repeatedly fit to bootstrapped subsets of the observations. On average each bagged tree makes use of around two-thirds of the observations.
- The remaining one-third of the observations not used to fit a given bagged tree are referred to as the ***out-of-bag*** (OOB) observations.
- We can predict the response for the ith observation using each of the trees in which that observation was OOB. This will yield around b/3 predictions for the ith observation, which we average.

Example of Random Forest

The table lists the details of five participants in a heart disease study, and a target feature RISK which describes their risk of heart disease.

Each patient is described in terms of four binary descriptive features

- EXERCISE, how regularly do they exercise
- SMOKER, do they smoke
- OBESE, are they overweight
- FAMILY, did any of their parents or siblings suffer from heart disease

ID	EXERCISE	SMOKER	OBESE	FAMILY	RISK
1	daily	false	false	yes	low
2	weekly	true	false	yes	high
3	daily	false	false	no	low
4	rarely	true	true	yes	high
5	rarely	true	true	no	high

Step 1. Generate bootstrap samples and random selection of m=2 features

ID	EXERCISE	FAMILY	RISK
1	daily	yes	low
2	weekly	yes	high
2	weekly	yes	high
5	rarely	no	high
5	rarely	no	high

Bootstrap Sample A

The entropy calculation for Sample A:

$$
\begin{aligned}
&H(\text{RISK}, BoostrapSampleA) \\
&= -\sum_{l \in \left\{\substack{low,\\ high}\right\}} P(\text{RISK} = l) \times log_2(P(\text{RISK} = l)) \\
&= -\left(\left(\frac{1}{5} \times log_2\left(\frac{1}{5}\right)\right) + \left(\frac{4}{5} \times log_2\left(\frac{4}{5}\right)\right)\right) \\
&= 0.7219\ bits
\end{aligned}
$$

ID	SMOKER	OBESE	RISK
1	false	false	low
2	true	false	high
2	true	false	high
4	true	true	high
5	true	true	high

Bootstrap Sample B

The entropy calculation for Sample B:

$$
\begin{aligned}
&H(\text{RISK}, BoostrapSampleB) \\
&= -\sum_{l \in \left\{\substack{low,\\ high}\right\}} P(\text{RISK} = l) \times log_2(P(\text{RISK} = l)) \\
&= -\left(\left(\frac{1}{5} \times log_2\left(\frac{1}{5}\right)\right) + \left(\frac{4}{5} \times log_2\left(\frac{4}{5}\right)\right)\right) \\
&= 0.7219\ bits
\end{aligned}
$$

ID	OBESE	FAMILY	RISK
1	false	yes	low
1	false	yes	low
2	false	yes	high
4	true	yes	high
5	true	no	high

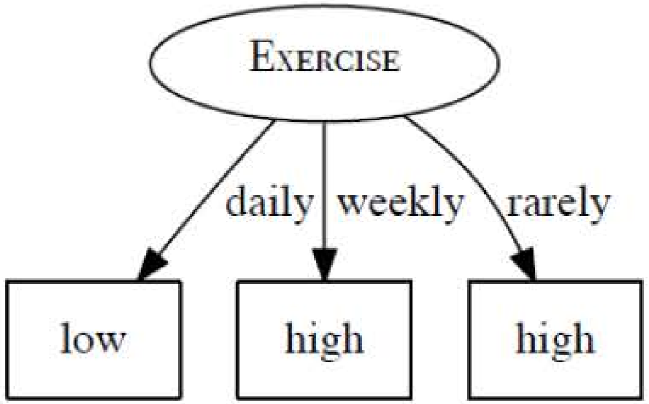
Bootstrap Sample C

The entropy calculation for Sample C:

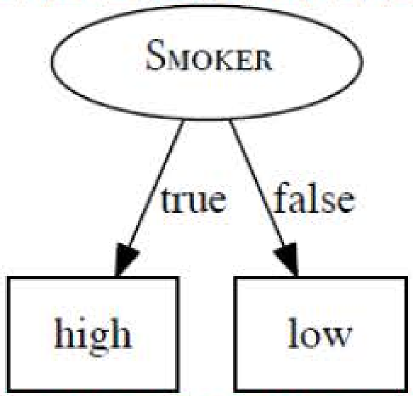
$$
\begin{aligned}
&H(\text{RISK}, BoostrapSampleC) \\
&= -\sum_{l \in \left\{\substack{low,\\ high}\right\}} P(\text{RISK} = l) \times log_2(P(\text{RISK} = l)) \\
&= -\left(\left(\frac{2}{5} \times log_2\left(\frac{2}{5}\right)\right) + \left(\frac{3}{5} \times log_2\left(\frac{3}{5}\right)\right)\right) \\
&= 0.9710\ bits
\end{aligned}
$$

Step 2. Grow a tree from each bootstrap sample

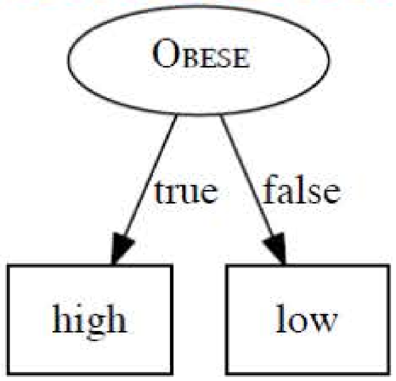
Split by Feature	Level	Instances	Partition Entropy	Rem.	Info. Gain
EXERCISE	*daily*	$\mathbf{d}_1$	0	0	0.7219
	weekly	$\mathbf{d}_2, \mathbf{d}_2$	0		
	rarely	$\mathbf{d}_5, \mathbf{d}_5,$	0		
FAMILY	*yes*	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_2$	0.9183	0.5510	0.1709
	no	$\mathbf{d}_5, \mathbf{d}_5$	0		



Split by Feature	Level	Instances	Partition Entropy	Rem.	Info. Gain
SMOKER	*true*	$\mathbf{d}_2, \mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_5$	0	0	0.7219
	false	$\mathbf{d}_1$	0		
OBESE	*true*	$\mathbf{d}_4, \mathbf{d}_5$	0	0.5510	0.1709
	false	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_2$	0.9183		



Split by Feature	Level	Instances	Partition Entropy	Rem.	Info. Gain
OBESE	*true*	$\mathbf{d}_4, \mathbf{d}_5$	0	0.5510	0.4200
	false	$\mathbf{d}_1, \mathbf{d}_1, \mathbf{d}_2$	0.9183		
FAMILY	*yes*	$\mathbf{d}_1, \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_4$	1.0	0.8	0.1709
	no	$\mathbf{d}_5$	0		



Step 3. Compute Out-of-Bag Error

- The observations not used to fit a given bagged tree are the ***out-of-bag*** (OOB) observations.

- ID=3, EXERCISE= daily, SMOKER=false, OBESE= false, FAMILY= no

Each of the trees in the ensemble will vote as follows:

- Tree 1: EXERCISE= daily → RISK=low
- Tree 2: SMOKER=false→ RISK=low
- Tree 3: OBESE= false → RISK=low

So, the majority vote is for RISK=low, same with the target RISK=low

Step 4. Make prediction

Assuming the random forest model you have created uses majority voting, what prediction will it return for the following query:

EXERCISE=rarely, SMOKER=false, OBESE=true, FAMILY=yes

Each of the trees in the ensemble will vote as follows:

- Tree 1: EXERCISE=rarely→ RISK=high
- Tree 2: SMOKER=false→ RISK=low
- Tree 3: OBESE=true→ RISK=high

So, the majority vote is for RISK=high

Algorithm 8.4 The random forests algorithm.

1. Given a training set (x_i, z_i), $i = 1, \ldots, n$, of patterns x_i and labels z_i. Specify the number of trees in the forest, B, and the number of random features to select, m.
2. For $b = 1, \ldots, B$,
   (a) Generate a bootstrap sample of size n by sampling with replacement from the training set; some patterns will be replicated, others will be omitted.
   (b) Design a decision tree classifier, $\eta_b(\boldsymbol{x})$ using the bootstrap sample as training data, randomly selecting at each node in the tree m variables to consider for splitting.
   (c) Classify the nonbootstrap patterns (the 'out-of-bag' data) using the classifier $\eta_b(\boldsymbol{x})$.
3. Assign $\boldsymbol{x}_i$ to the class most represented by the classifiers $\eta_{b'}(\boldsymbol{x})$, where b' refers to the bootstrap samples that do not contain $\boldsymbol{x}_i$.

Andrew R.Webb , Keith D. Copsey, Statistical Pattern Recognition, 3rd ed. 2011

Summary

- Decision trees are simple and interpretable models for regression and classification.
- However they are often not competitive with other methods in terms of prediction accuracy.
- Bagging and random forests are good methods for improving the prediction accuracy of trees. They work by growing many trees on the training data and then combining the prediction of the resulting ensemble of trees.
- Random forests is one of the state-of-the-art methods for supervised learning. However results can be difficult to interpret.

Additional Tutorial (StatQuest)

Decision tree:
https://www.youtube.com/watch?v=J4Wdy0Wc_xQ

Random forest:
Part I https://www.youtube.com/watch?v=J4Wdy0Wc_xQ&t=123s
Part II https://www.youtube.com/watch?v=sQ870aTKqiM

AdaBoost:
https://www.youtube.com/watch?v=LsK-xG1cLYA

Gradient Boost
Part I https://www.youtube.com/watch?v=3CC4N4z3GJc&t=50s
Part II https://www.youtube.com/watch?v=2xudPOBz-vs
Part III https://www.youtube.com/watch?v=jxuNLH5dXCs

Algorithm 8.2 The Adaboost algorithm.

1. Initialise the weights $w_i = 1/n$, $i = 1, \ldots, n$.
2. For $t = 1, \ldots, T$, (T is the number of boosting rounds)
   (a) Construct a classifier $\eta_t(\boldsymbol{x})$ from the training data with weights w_i, $i = 1, \ldots, n$.
   (b) Calculate e_t as the sum of the weights w_i corresponding to misclassified patterns.
   (c) If $e_t > 0.5$ or $e_t = 0$ then terminate the procedure, otherwise set $w_i = w_i(1 - e_t)/e_t$ for the misclassified patterns and renormalise the weights so that they sum to unity.
3. For a two-class classifier, in which $\eta_t(\boldsymbol{x}) = 1$ implies $\boldsymbol{x} \in \omega_1$ and $\eta_t(\boldsymbol{x}) = -1$ implies $\boldsymbol{x} \in \omega_2$, form a weighted sum of the classifiers, η_t,

$$\hat{\eta} = \sum_{t=1}^{T} \log\left(\frac{1 - e_t}{e_t}\right) \eta_t(\boldsymbol{x})$$

and assign $\boldsymbol{x}$ to ω_1 if $\hat{\eta} > 0$.

Andrew R.Webb , Keith D. Copsey, Statistical Pattern Recognition, 3rd ed. 2011