

Statistical Learning (AMI22T)

Mar. 30, 2020



HÖGSKOLAN
DALARNA

Learning outcomes

Upon completion of this course, the students shall be able to:

- Select the correct statistical models, and methods for a data analysis problem in the real world based on reasoned argument, especially when the underlying data generating mechanism is unknown.
- Apply various supervised and unsupervised statistical learning algorithms in a range of real-world problems.
- Evaluate, and optimise the performances of the learning models and algorithms, and communicate the expected accuracy of the model/algorithm.
- Combine several models to achieve higher predictive accuracy.

Assessment tasks

Table 1 Points distribution among of the assessment tasks

Task	Points
Home Exercise (3)	60 (20×3)
MCQ Exam (1)	30
Oral Exam	10

Grading

- All the points from home exercises, and final exam will be summed up to determine final grade.
- The course is graded in U-VG scale.
- U : <50, G: 50> and VG: >75
- Failure to return the solutions of home exercises before deadline leads to 0 point.
- You are entitled to more than one re-take on MCQ and oral exam.
- The form assessment is valid only for Spring 202.

More on home exercises

- Should be done independently.
- Provided with a cover page stating your name, e-mail, etc.
- You should return your solution via Learn.
- Length of exercises: not more than 12 pages.
- Your solution should be structured as: Introduction-Statistical Methods-Results-Discussion. You may add a conclusion section too.
- You should be able to motivate your methods and justify your conclusion.
- R codes can be supplied in a separate document or in an appendix.

What is a Statistical Learning?

- Statistical Learning is a subfield in Statistics, focused on supervised and unsupervised modeling and prediction.
- Statistical Learning refers to a vast set of tools for understanding data. These tools can be classified as supervised and unsupervised learning.

Input and output variables

- Assume a relationship between a quantitative variable Y , and some predictors $X=(X_1, X_2, \dots, X_p)$ gives as

$$Y = f(X) + \epsilon$$

- We call $X=(X_1, X_2, \dots, X_p)$ as the predictor, independent variables, features, or input variables.
- We call Y as the dependent, response, or output variable.
- If our target is to predict Y using $\hat{Y} = \hat{f}(X)$ then we call it a prediction problem.
- If we are interested in answering the following questions
 1. Which predictors are associated with the response?
 2. What is the relationship between the response and predictor?
 3. How can the relationship between Y and X be formulated, e.g. using linear equation or using more complicated function?

Then we are dealing with inference.



Supervised vs. Unsupervised Learning

- Supervised learning refers to connecting response to the predictor variables. Example: Linear regression, logistic regression, SVM, etc.
- Unsupervised learning refers to understanding the relationship between variables (without classifying them as input and output), or between observations. Example: Principal component analysis, cluster analysis, etc.



Regression Vs. Classification

- Problems with quantitative response are referred to as regression.
- Problems involving qualitative response are often referred to as classification problem.

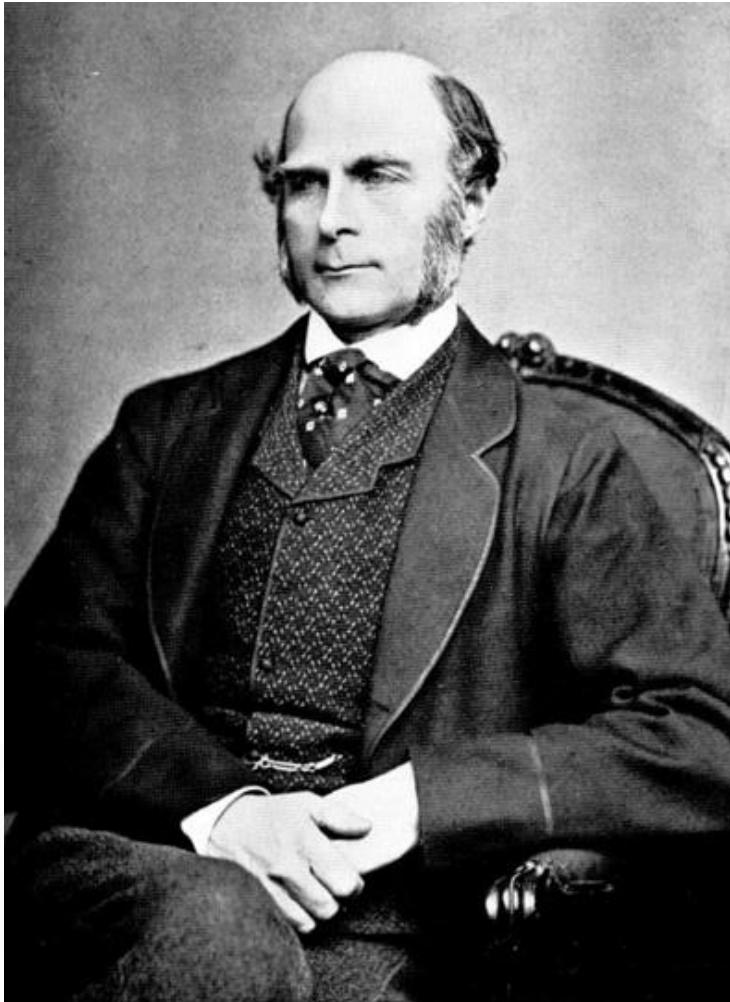


Gauss and Legendre: inventors of least square



KOLAN
NA

F. Galton introduces the term, “regression”



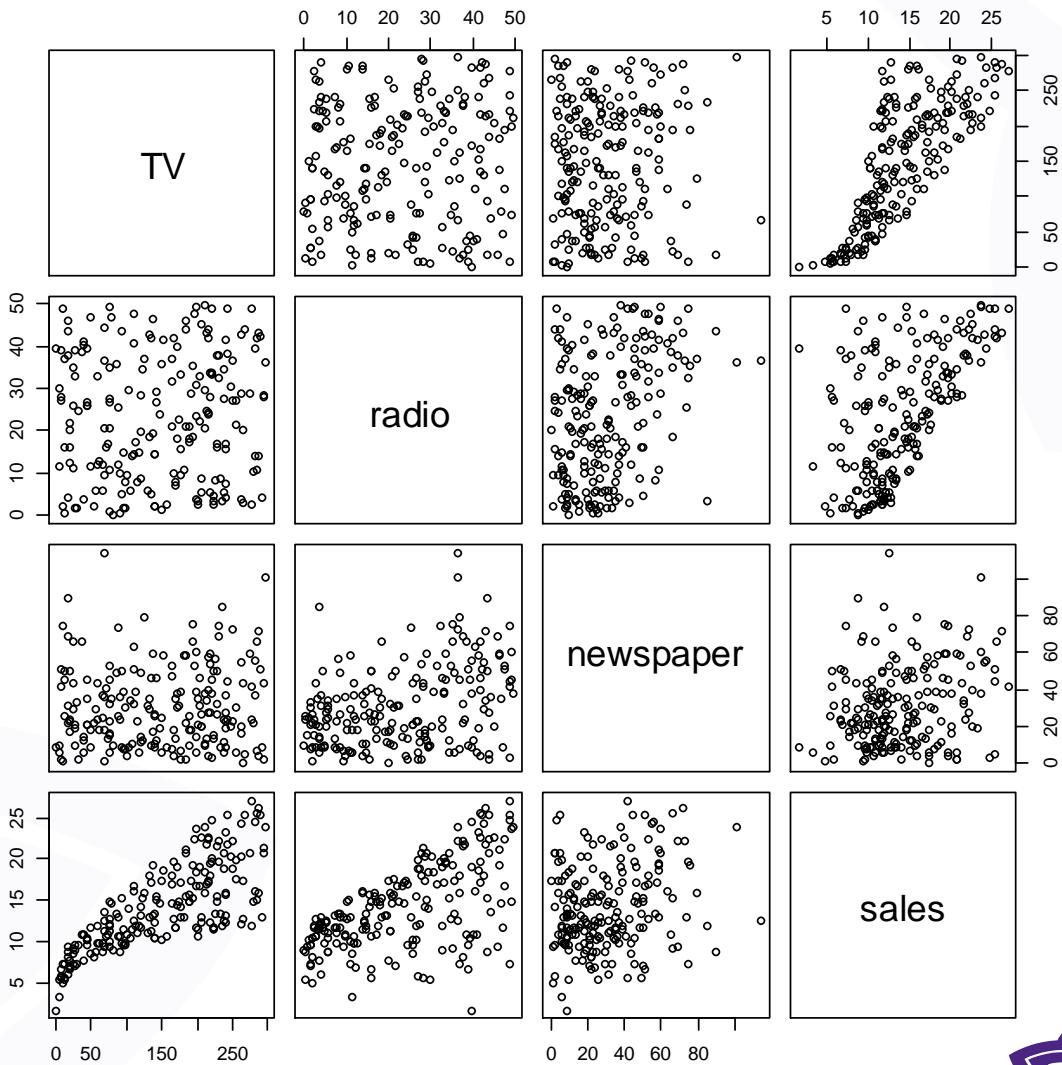
Content of the rest of the lecture

- Formulation of a linear model.
- Model assumptions.
- Linear model in matrix notation.
- Least square estimation of a linear model.
- Properties of the least square estimator.
- Explanation of model parameters.
- Assessing goodness of fit.
- Inference with linear model.
- See pp. 1-7, 24 and Appendix A in Ulf Olsson's (2002) book.

Advertising data set

- Advertising data set gives sales of a product in 200 different markets along with advertising budget in three media: TV, newspaper, and radio.
- With this data set we might want to answer the following questions
 - Is there any relationship between advertising budget and sales?
 - How strong is the relationship between advertising budget and sales?
 - Which media contribute to sales?
 - How accurately can we estimate the effect of each medium on sales?
 - How accurately can we predict future sales?

Scatterplot of Advertising data set



Linear regression

- To answer the research questions, we may set up linear regression, e.g.

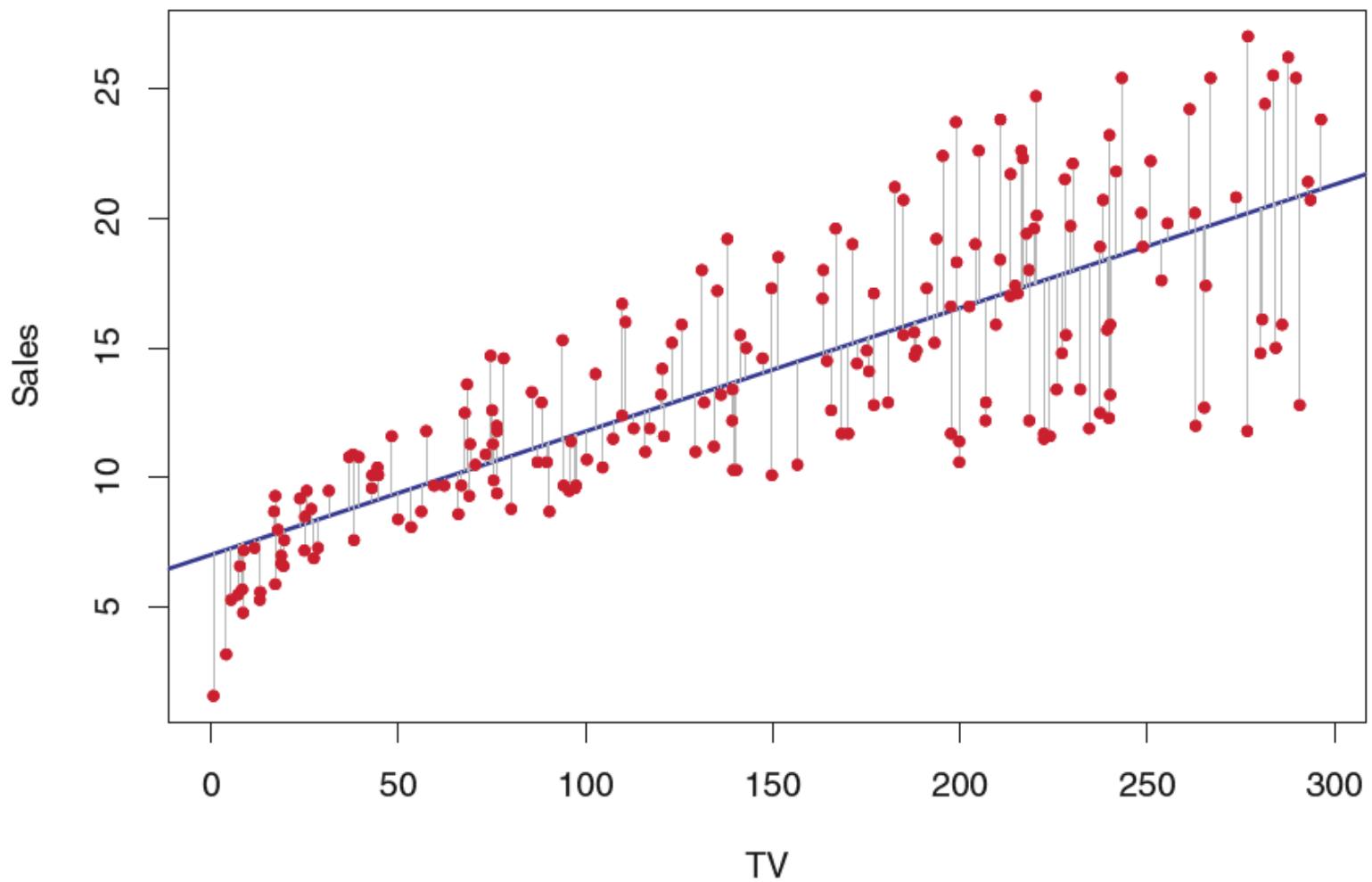
$$\text{Sales} \approx \beta_0 + \beta_1 \times TV$$

$$\text{Or, } Y = \beta_0 + \beta_1 X + \varepsilon$$

- Here, β_0 and β_1 are unknown constant, or coefficients, parameters.
- We need to find some estimate $\hat{\beta}_0$, and $\hat{\beta}_1$ using training data.



Ordinary Least-square (OLS) estimation



Ordinary least-square estimation (Cont.)

- Let $y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$, and $e_i = y_i - \hat{\beta}_0 + \hat{\beta}_1 x_i$
- Residual sum of square is given by

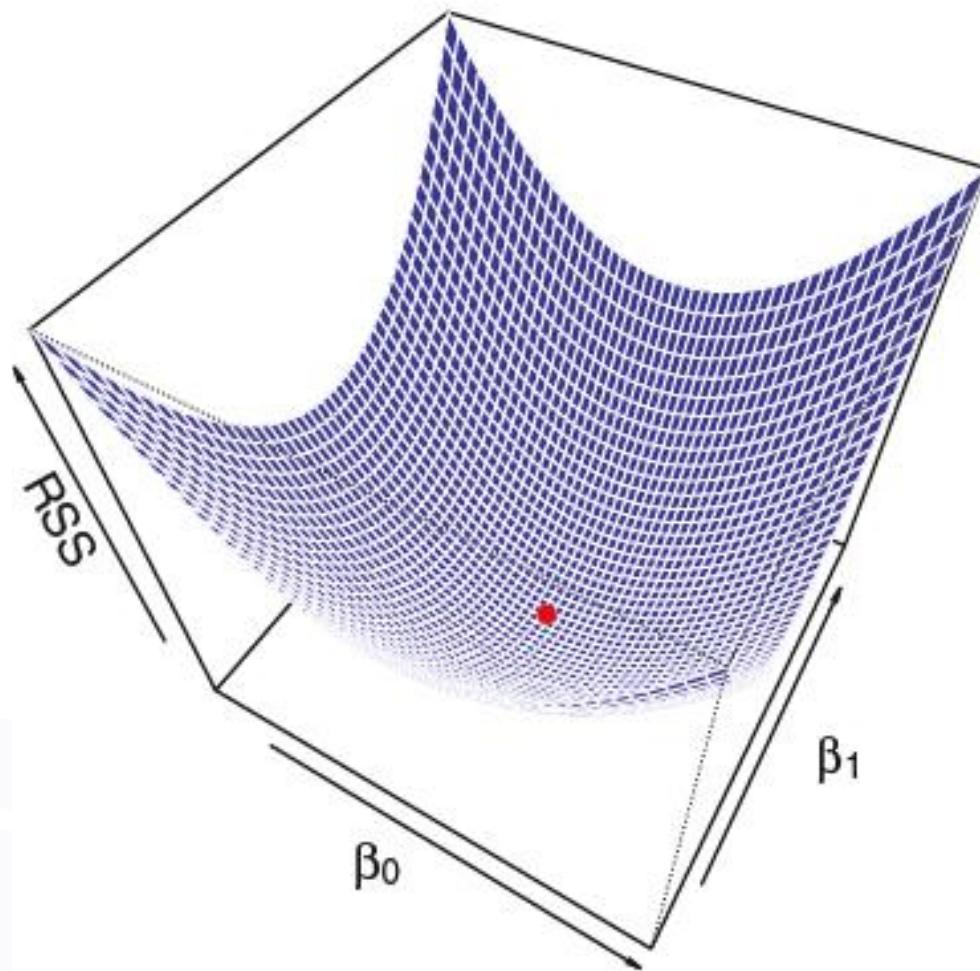
$$RSS = e_1^2 + e_2^2 + \dots + e_n^2 = \sum_i (y_i - \hat{\beta}_0 + \hat{\beta}_1 x_i)^2$$

- Minimizing RSS we have

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

and $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$

Plot of the least-square objective function



Some properties of OLS estimator

Under a set of mild assumptions, OLS estimator enjoys the following properties

- OLS estimator is unbiased, i.e. $E(\hat{\beta}_j) = \beta_j$; $j=1, 2$.
- If $\tilde{\beta}_j \neq \hat{\beta}_j$, is another linear unbiased estimator of β_j then $\text{Var}(\tilde{\beta}_j) > \text{Var}(\hat{\beta}_j)$
- $\lim_{n \rightarrow \infty} \text{Var}(\hat{\beta}_j) = 0$
- In large sample, $\hat{\beta}_j \sim N(\beta_j, \text{Var}(\hat{\beta}_j))$



Uncertainties in OLS estimator

$$\text{SE}(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right], \quad \text{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

Where $\text{Var}(\varepsilon_i) = \sigma^2$, and it is replaced by its unbiased estimator $\hat{\sigma}^2 = \frac{\text{RSS}}{n-2}$

A 95% confidence interval of $\hat{\beta}_j$ is given by
$$\hat{\beta}_j \pm 1.96 \times \text{SE}(\hat{\beta}_j)$$

when “n” is large.



Hypothesis test about coefficients

- Assume we want to test the null hypothesis

H_0 : *There is no relationship between TV add and sales*

Against

H_1 : *There is some relationship between TV add and sales*

Mathematically this corresponds to

$H_0: \beta_1 = 0$ vs. $H_1: \beta_1 \neq 0$

The above H_0 is tested using the test statistic

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)} \sim t_{n-2}$$



Sales and TV add examples

```
> summary(lm(sales~TV,data=add))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.032594	0.457843	15.36	<2e-16 ***
TV	0.047537	0.002691	17.67	<2e-16 ***
<hr/>				

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.259 on 198 degrees of freedom

Multiple R-squared: 0.6119, Adjusted R-squared: 0.6099

F-statistic: 312.1 on 1 and 198 DF, p-value: < 2.2e-16



Extension to multiple linear regression

- So far, we focused o relationship between sales, and TV adds. But, what about other adds?
- For this, we can formulate a multiple linear regression

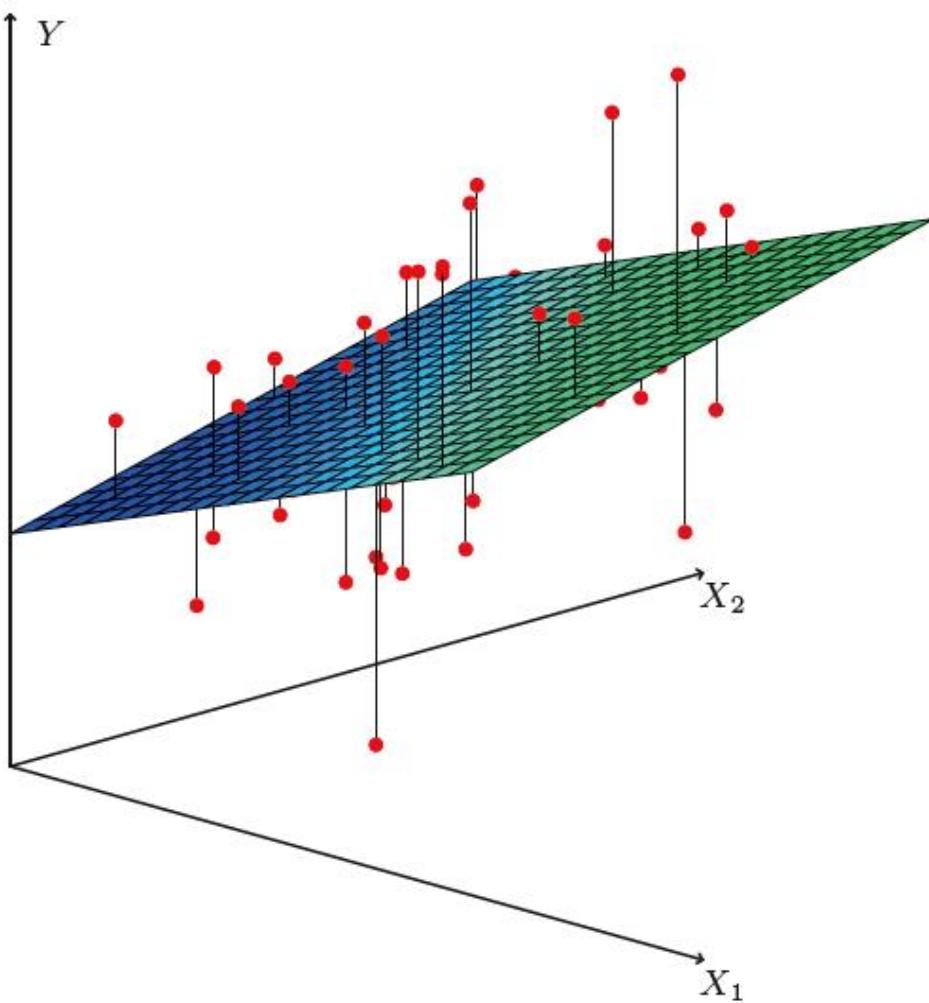
$$sales = \beta_0 + \beta_1 \times TV + \beta_2 \times radio + \beta_3 \times newspaper + \varepsilon$$

- This model can be estimated by minimizing the following RSS

$$RSS = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{1i} - \cdots - \beta_3 x_{3i})^2$$



Plot of a three variable regression



OLS estimator of multiple linear regression

- Denote $y = (y_1, y_2, \dots, y_n)^T$, $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)^T$, $p < n$, and

$$X = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{p,1} \\ 1 & x_{1,2} & \vdots & x_{p,2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1,n} & \dots & x_{p,n} \end{pmatrix}$$

- Then the RSS is given by: $RSS = (y - X\hat{\beta})^T(y - X\hat{\beta})$
- Minimizing RSS the OLS estimator $\hat{\beta}$ is given by
$$\hat{\beta} = (X^T X)^{-1} X^T y$$
- Also $Var(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$



Is there any relationship between response and the predictors

- We test this with the following hypothesis

$$H_0: \beta_1 = \beta_2 = \dots = \beta_p = 0 \text{ vs } H_a: \beta_j \neq 0 \text{ for some } j$$

- We test this hypothesis using the so called F-test (also called ANOVA)

$$F = \frac{(TSS - RSS)/p}{RSS/(n - p - 1)} \sim F_{p,n-p-1}$$

where $TSS = \sum_{i=1}^n (y_i - \bar{y})^2$



Predictor variable selection

- We can use:
 - Forward selection: Start with null model then gradually add variable, depending on some model fit criteria.
 - Backward selection: Start with largest possible model then gradually drop (insignificant) variables.
 - Mixed selection: Use both forward and backward method, and choose the one turns out the best.
- Use AIC, BIC, C_p , etc. for model selection
- In this course, we will use backward selection method, relying on t-test or F-test at a liberal significance level e.g. 20% or 25%.

Advertising Example (cont.)

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.938889	0.311908	9.422	<2e-16 ***
TV	0.045765	0.001395	32.809	<2e-16 ***
radio	0.188530	0.008611	21.893	<2e-16 ***
newspaper	-0.001037	0.005871	-0.177	0.86

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.686 on 196 degrees of freedom

Multiple R-squared: 0.8972, Adjusted R-squared: 0.8956

F-statistic: 570.3 on 3 and 196 DF, p-value: < 2.2e-16

Contents of the next lecture

- R^2 statistic
- Regression with categorical predictor.
- Dummy coding and explanation
- Extension of linear model
- Potential problems with linear model
- Conceptual exercises

Statistical Learning (AMI22T)

Mar. 31 , 2020



HÖGSKOLAN
DALARNA

Contents of Lecture 2

- Goodness of fit
- Regression with categorical variable.
- Dummy coding and explanation
- Model diagnostics
- Model building and model selection
(example)

Model fit statistics

- Two most common descriptive measure of model fit are: Residual Standard Error (RSE), and R^2 statistic.
- $RSE = \sqrt{\frac{RSS}{n-p-1}} = \sqrt{\frac{\sum_{i=1}^n e_i^2}{n-p-1}}$; $0 \leq RSE < \infty$
- $R^2 = 1 - \frac{RSS}{TSS}$, $TSS = \sum_i (y_i - \bar{y})^2$; $0 \leq R^2 \leq 1$
- In general $R^2 = (\text{Cor}(y_i, \hat{y}))^2$, and for a two-variable regression $R^2 = r^2 = (\text{Cor}(x, y))^2$
- Adjusting R^2 for number of covariates give

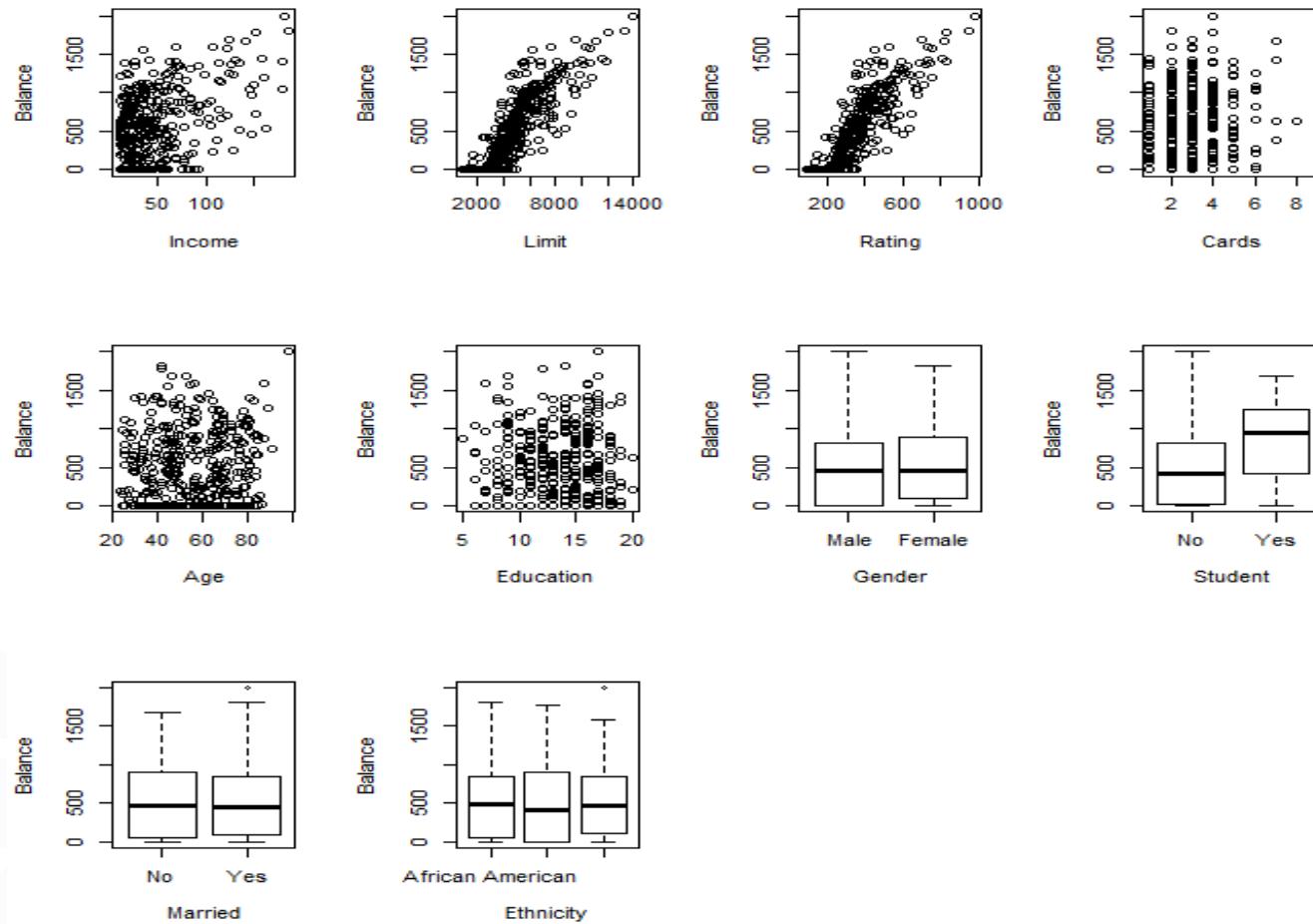
$$R^2_{Adj.} = 1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}$$



Regression with categorical variable

- For the fitting of a linear regression model, we need all the variables to be numerical.
- If, we have some categorical predictors in the data, we need to convert them into numerical.
- The most popular way of doing this, is known as dummy coding of the categorical variables.
- There are other solutions, too.

Balance vs. predictors (data: Credits)



Categorical variable with two levels

- Suppose we want to examine whether the average credit card balance differs between gender.
- Because the categorical variable “Gender” has tow levels (Male and Female) We can create new dummy variable as:

$$x_i = \begin{cases} 1, & \text{if } Gender_i = female \\ 0, & \text{if } Gender_i = male \end{cases}$$

- Now, instead of Gender, we can use x as a predictor in a linear model.



Categorical variable with two levels (cont.)

- Linear regression model of balance on x is given as

$Balance_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, with $\varepsilon_i \sim N(0, \sigma^2)$ being an irreducible error term

- This implies that

$$E(Balance_i | Gender = Male) = \beta_0 \text{ and}$$

$$E(Balance_i | Gender = Female) = \beta_0 + \beta_1$$

- In other words, β_1 gives the average difference between male and female.



Analysis of credit cards balance

```
> aggregate(Balance~Gender,FUN=mean,data=Credit)
```

```
  Gender Balance
```

```
1 Male 509.8031
```

```
2 Female 529.5362
```

```
> summary(lm(Balance~Gender,data=Credit))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	509.80	33.13	15.389	<2e-16 ***
GenderFemale	19.73	46.05	0.429	0.669

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 460.2 on 398 degrees of freedom

Multiple R-squared: 0.0004611, Adjusted R-squared: -0.00205

F-statistic: 0.1836 on 1 and 398 DF, p-value: 0.6685



Categorical variable with more than two levels

- In Credit data set, think about Ethnicity variable which has three levels.
- In this case we can create following two dummy variable as:

$$x_{i1} = \begin{cases} 1, & \text{if } \text{Ethnicity}_i = \text{Asian} \\ 0, & \text{if } \text{Ethnicity}_i \neq \text{Asian} \end{cases}$$
$$x_{i,1} = \begin{cases} 1, & \text{if } \text{Ethnicity}_i = \text{Caucasian} \\ 0, & \text{if } \text{Ethnicity}_i \neq \text{Caucasian} \end{cases}$$

- Now, we can use x_1 and x_2 as two predictors in a linear model.
- We cannot use 3 dummy variables corresponding to 3 levels.

Categorical variable with more than two levels (cont.)

- Denoting $y = Balance$, we can write the regression model of y on Ethnicity as

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$$

- This implies

$$y_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i, & \text{if } \text{Ethnicity}_i = \text{Asian} \\ \beta_0 + \beta_2 + \varepsilon_i, & \text{if } \text{Ethnicity}_i = \text{Caucasian} \\ \beta_0 + \varepsilon_i, & \text{if } \text{Ethnicity}_i = \text{African American} \end{cases}$$

- Therefore, we can interpret β_0 as the average balance for African American, β_1 is the difference in balance between African American and Asian, and β_2 is the difference between Caucasian and African American.



Regression of Balance on Ethnicity

Call:

```
lm(formula = Balance ~ Ethnicity, data = Credit)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	531.00	46.32	11.464	<2e-16 ***
EthnicityAsian	-18.69	65.02	-0.287	0.774
EthnicityCaucasian	-12.50	56.68	-0.221	0.826
<hr/>				

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				

Residual standard error: 460.9 on 397 degrees of freedom

Multiple R-squared: 0.0002188, Adjusted R-squared: -0.004818

F-statistic: 0.04344 on 2 and 397 DF, p-value: 0.9575



Extension of linear models

- We can extend linear model to deal with non-linearity, under the same estimation procedure.
- In Advertising data, sales effect of adds in one media might depend on how much is spent for advertising in another media. This gives

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,1} x_{i,2} + \epsilon_i$$

- We call the term $\beta_3 x_{i,1} x_{i,2}$ interaction effect.
- In Credit data, effect of income on balance might be different depending on whether the individual is a student or not. This gives

$$y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \beta_3 X_{i,1} X_{i,2} + \epsilon_i$$

where $X_{i,1} = \text{Income}_i$, and $X_{i,2} = I(\text{Student}_i = \text{Yes})$

Credits balance on Income and Student

Call:

```
lm(formula = Balance ~ Income * Student, data = Credit)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	200.6232	33.6984	5.953	5.79e-09 ***
Income	6.2182	0.5921	10.502	< 2e-16 ***
StudentYes	476.6758	104.3512	4.568	6.59e-06 ***
Income:StudentYes	-1.9992	1.7313	-1.155	0.249

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

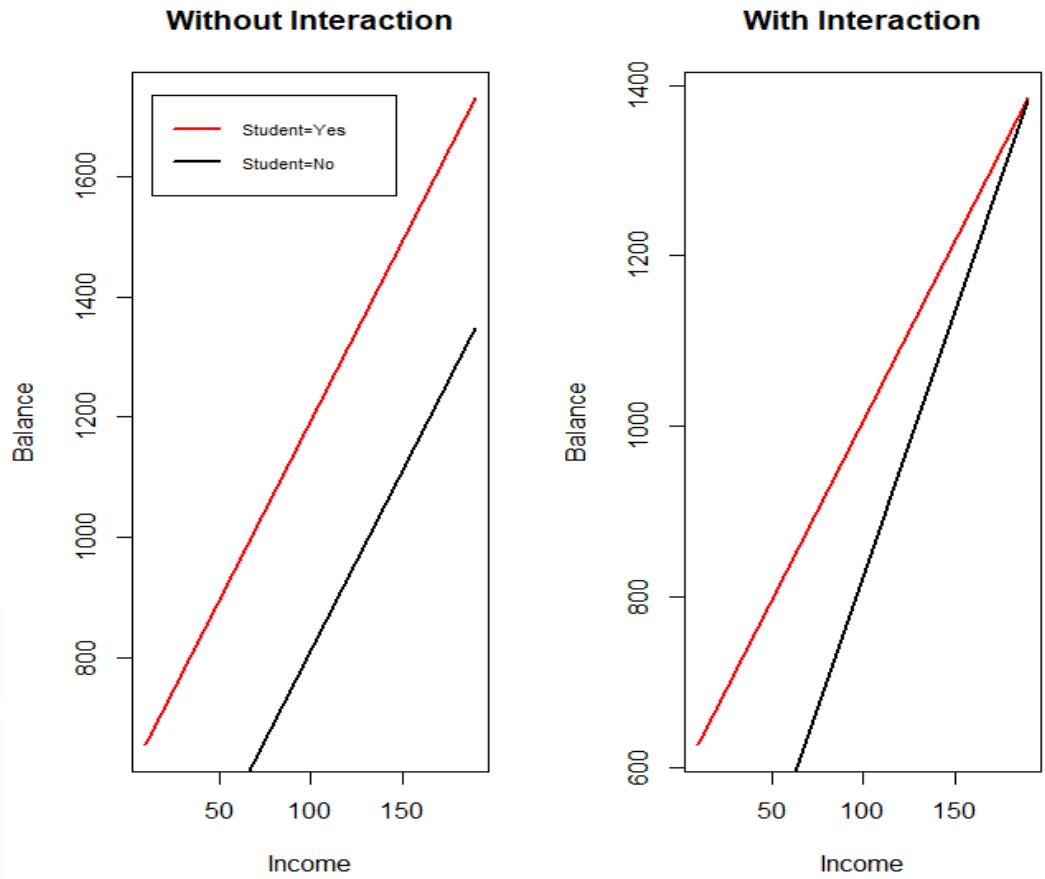
Residual standard error: 391.6 on 396 degrees of freedom

Multiple R-squared: 0.2799, Adjusted R-squared: 0.2744

F-statistic: 51.3 on 3 and 396 DF, p-value: < 2.2e-16



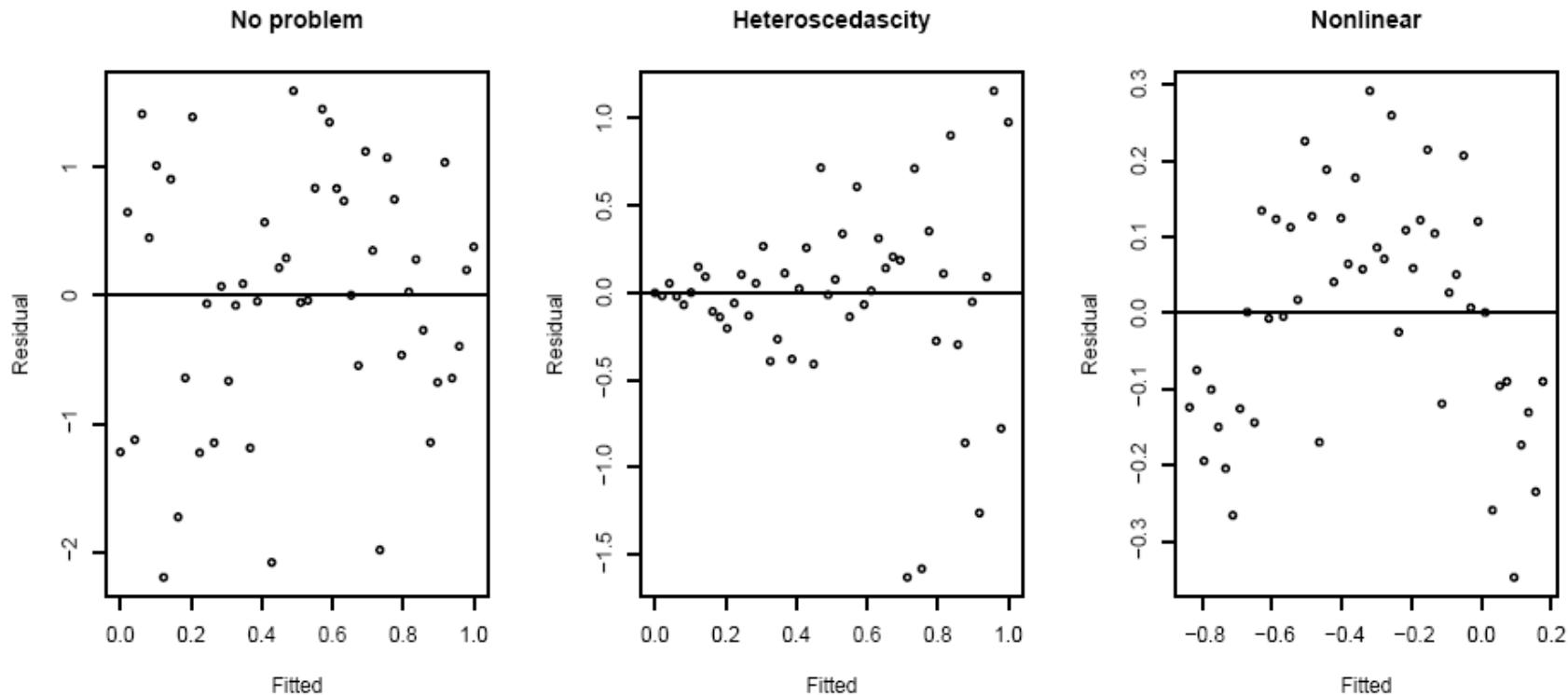
Visualizing interaction effect: Income and Credit balance example



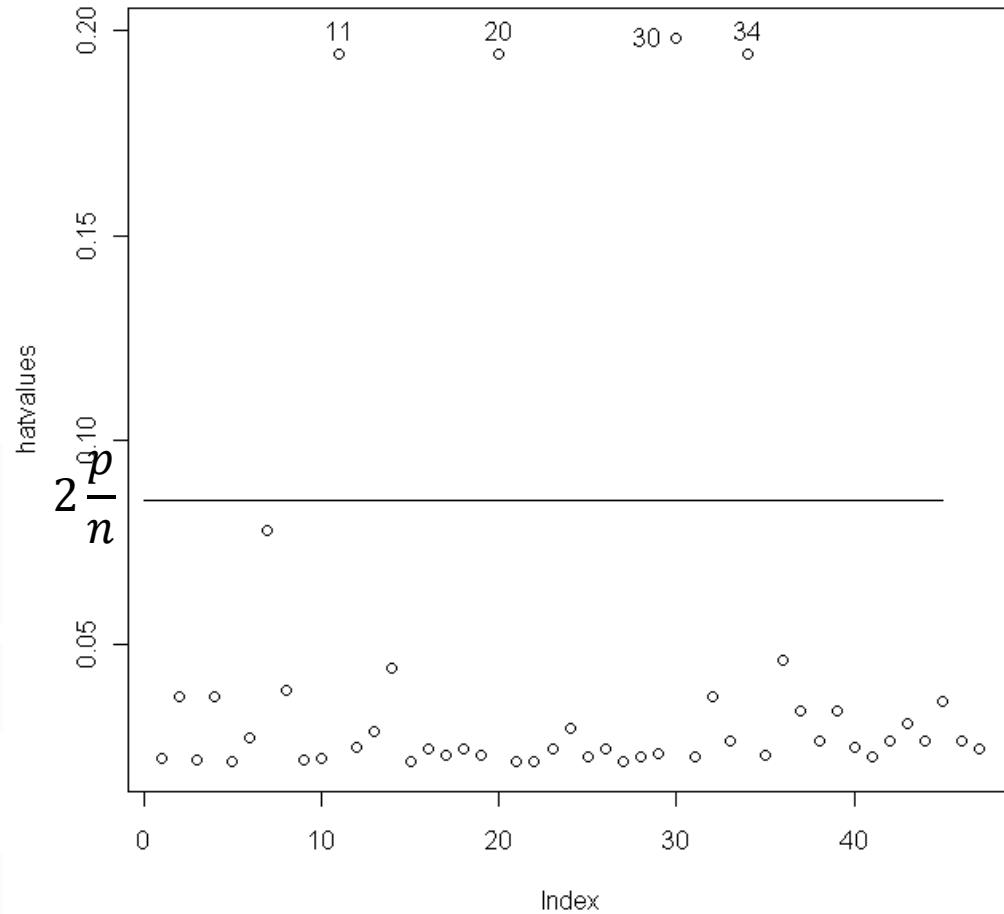
What might go wrong with OLS?

- Non-linear relationship between response and predictor: Parameter estimates are not meaningful.
- Errors have unequal variance (heteroscedasticity) or are correlated with each other (autocorrelation): OLS estimator is inefficient, and standard error estimates are wrong.
- Some of the X's are highly correlated (collinearity): Numerically unstable or not estimable.
- Outliers in the data: Estimates are highly influenced by a few extreme values.
- Errors are correlated with some X's or X's are subject to measurement error: Biased estimate...

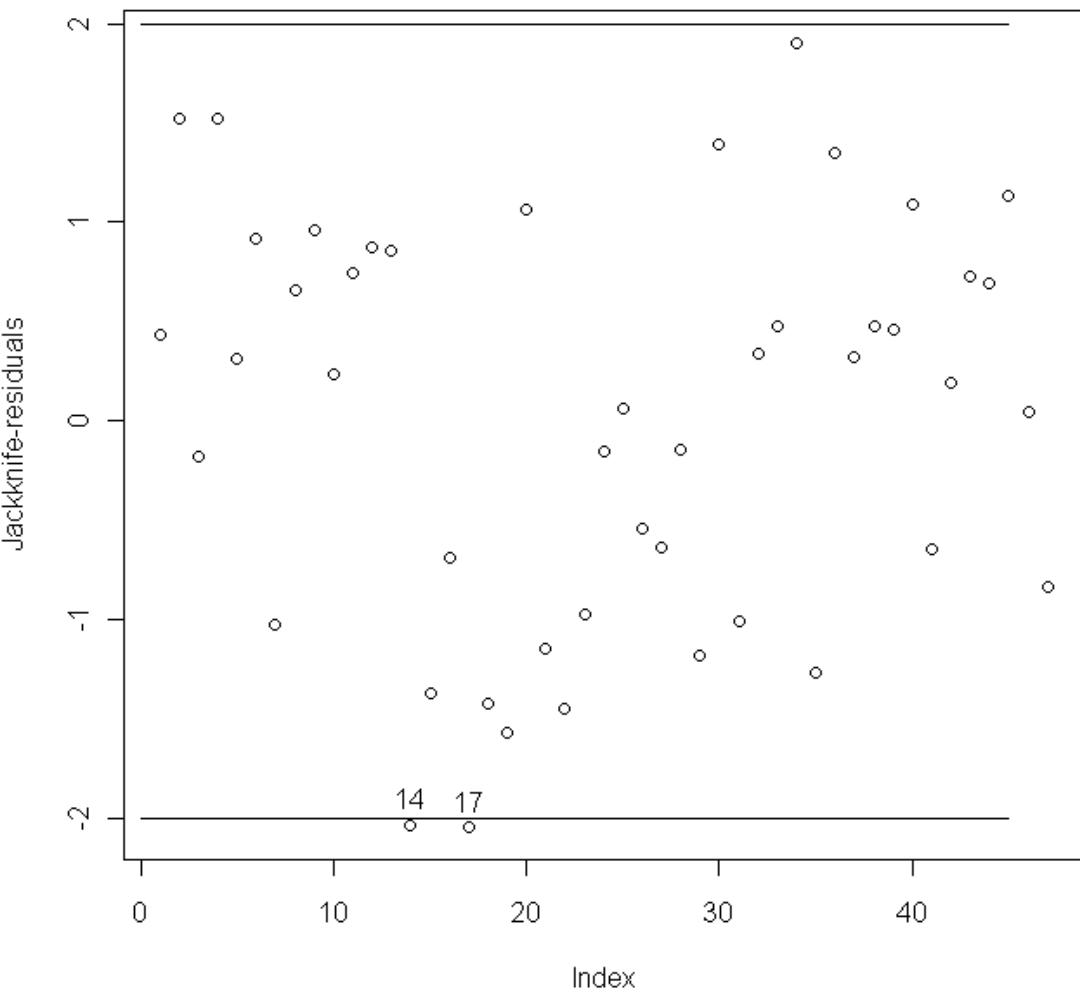
Plot of residuals vs. fitted values



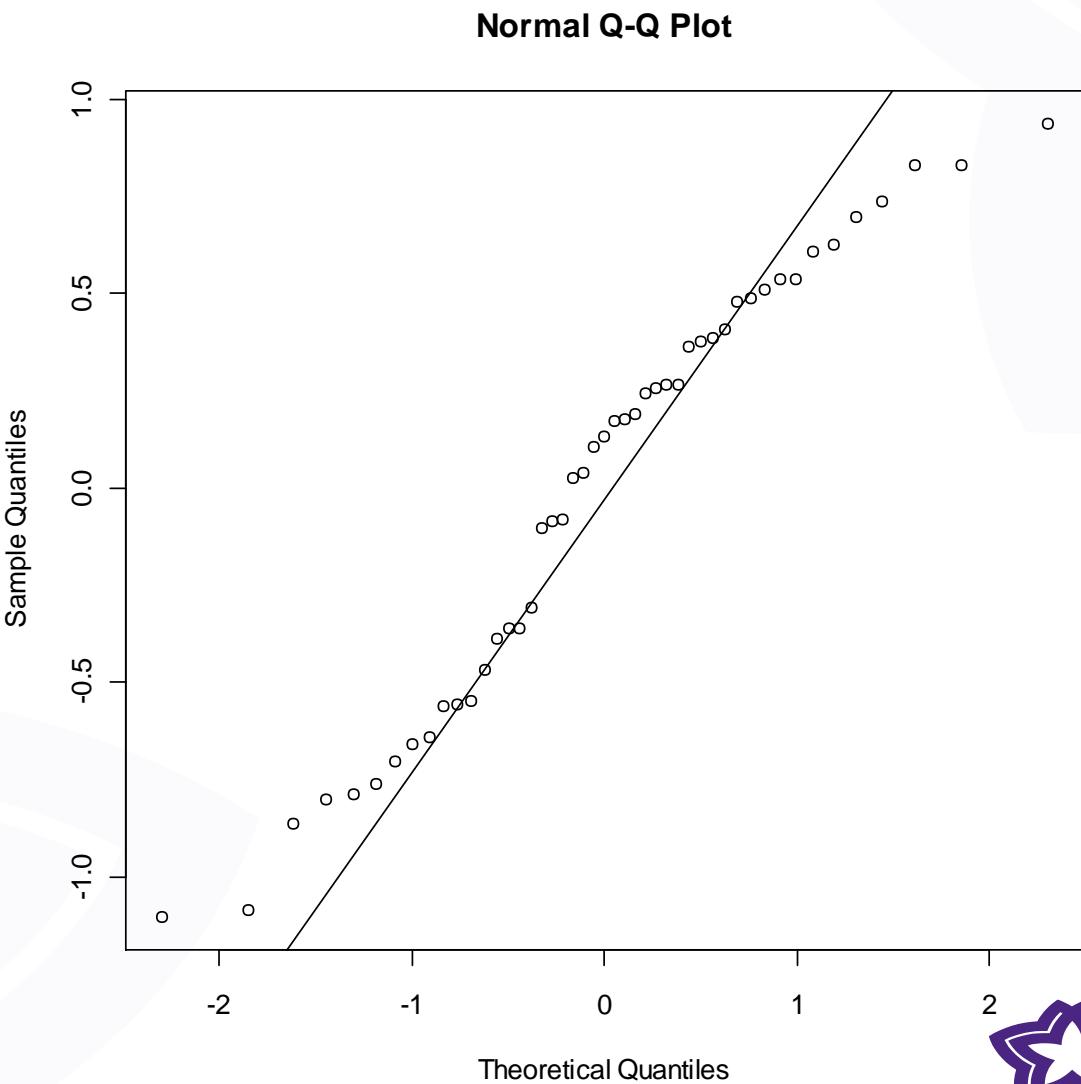
Plotting the hat values for influential observation (high leverage points)



Standardized and Jack-knife residual plot



Q-Q plot of the residuals



Other diagnostic measures

- Cook's distance: $D_i \geq \frac{4}{n-k-1}$ is an indication of outlier.
- Plot |residual| against fitted values.
- Plot residuals against their lagged values.
- Plot residual against those variables which are not included in the model.
- Check component+residuals plots. Any curvature show possible non-linearity in X.
- Check Box-Cox plot. This would suggest if some power transformation the response variable can help achieving better fit.
- Check VIF statistic. As a rule of thumb, $VIF > 10$ indicates collinearity.



Remarks on model diagnostics

- An abnormal value is not necessarily a wrong value. Removal of correct extreme values might lead to a wrong conclusion.
- Transformation of variables may remove outliers.
- In presence of many outliers, big ones might hide (mask) the small outliers.
- Robust regression (e.g. M-estimator) methods are often very useful in detecting outliers.
- Use more than one measures for outlier detection.
- You should mention in your analysis if you took any action about the outliers.

Step-wise (backward) model selection

- We strive for a simple model that captures all relevant information from the data.
- Start with all relevant main effects and interactions
- If an interaction is included, corresponding main effects are also included even though it is not significant.
- Remove a lower order term after all of its higher order terms of it are removed.
- Any covariate that does not have any significant effect should be deleted.
- Conventional 5% level of significance may be too strict for model selection. Use 20-25% level.
- AIC can be used for model selection.
- Any relevant subject-matter point should be considered.

(see Ulf Olsson (2002), Generalized linear models: An applied approach)



HÖGSKOLAN
DALARNA

Some quotations about model selection

- All models are wrong but some are useful (G. Box).
- Any fool can make things bigger, more complex, and more violent. It takes a touch of genius and a lot of courage to move in the opposite direction. (A. Einstein).

Example: Model selection

```
library(MASS); library(car); library(faraway)
data(gala); ?gala
l1<-lm(I(Species)~poly(Elevation,3)+log(Area)+
poly(Nearest,3)+Scruz+poly(Adjacent,3)+I(Area/Nearest),data=gala)
boxcox(l1)
l2<-lm(I((Species)^(1/3))~poly(Elevation,3)+log(Area)+
poly(Nearest,3)+Scruz+poly(Adjacent,3)+I(Area/Nearest),data=gala)
summary(l2); boxcox(l2)
l3<-update(l2,.~.-poly(Adjacent,3)+poly(Adjacent,2)) #drop ins. hig. term
l3<-update(l3,.~.-poly(Nearest,3)+poly(Nearest,2))
l3<-update(l3,.~.-poly(Nearest,2)+Nearest)
l3<-update(l3,.~.-Scruz)
summary(l3)
```

Exercises (Chapter 3)

- Check exercises: 3, 4, and 7
- For Lab (April 1), check: 8, 9, 13, and 14

Statistical Learning (AMI22T)

April. 06 , 2020

Classification: Logistic Regression



HÖGSKOLAN
DALARNA

Contents of Lecture 3

- Review of Exercises from Chapter 3
- Regression and Classification.
- Logistic regression for inference and classification
- Estimation of logistic regression
- Interpretation of model parameters
- Prediction using logistic regression
- Extension of logistic regression
- What might go wrong with logistic regression

- So far, we have dealt with regression models that deals with quantitative response.
- Problems involving qualitative response is referred to as classification problem.
- Suppose we want to predict which customer will default on credit card debt, based on income, and whether s/he is a student.
- Here the response variable “default” is qualitative with two levels “Yes” and “No”.
- In this case, we can model $P(\text{default}=\text{Yes} | X)$.

Why can't we use linear regression for classification

- It seems we can dummy code default and use a simple linear regression.
- Even though a linear model is mathematically feasible, in this case, it's predicted values are not guaranteed to lie between 0, and 1.
- If response variable contains more than two levels, dummy (or numerical) coding turns out a difficult task, as the results might differ based on actual numerical coding.

Logistic regression

- Let us denote $P(Y = 1|X) = p(X)$.
- A linear regression would model $p(X) = \alpha + \beta X$
- In logistic regression we specify the relationship as

$$p(X) = \frac{e^{\alpha+\beta X}}{1 + e^{\alpha+\beta X}}$$

- After a bit of algebraic manipulation we get

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \alpha + \beta X$$

- The quantity $\frac{p(X)}{1-p(X)}$ is called odds

- The ratio $\frac{\frac{p(X_1)}{1-p(X_1)}}{\frac{p(X_2)}{1-p(X_2)}}$ is called odds-ratio.



Some notes about OR

- Let $p_1 = 0.01$, and $p_2 = 0.001$ then $p_1 - p_2 = 0.009$ (too small!)
- Let $p_1 = 0.41$, and $p_2 = 0.401$ then $p_1 - p_2 = 0.009$ (same difference!)
- OR for the first case is $\frac{0.01/0.99}{0.001/0.999} = 10.09$,
and for the second case is $\frac{0.41/0.59}{0.401/0.599} = 1.04$
- For logistic regression, e^β gives odds-ratio for unit change in x
- What does $OR=1$ imply?

Estimation of logistic regression

- Logistic regression is not estimated using least-square method.
- Logistic regression is estimated using Maximum Likelihood (ML) method which minimizes the following objective function

$$-l = - \sum_{i:y_i=1} \log(p(X_i)) - \sum_{i:y_i \neq 0} \log(1 - p(X_i))$$

- At the end of the day, the ML estimation turns out to be an iterative weighted least square.
- In matter of fact, a wide class of model can be fitted with the same algorithm, and this algorithm implemented in `glm` function in R.



Analysis of Default data set

```
summary(glm(default~balance,data=Default,family=binomial))
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.065e+01	3.612e-01	-29.49	<2e-16 ***
balance	5.499e-03	2.204e-04	24.95	<2e-16 ***
<hr/>				

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2920.6 on 9999 degrees of freedom

Residual deviance: 1596.5 on 9998 degrees of freedom

AIC: 1600.5

Extension of logistic regression

- We can extend logistic regression to accommodate any number of features (p), as long as $p < n$.
- Qualitative covariates can be added using the same dummy coding technique as in linear model.
- Coefficient estimates have OR interpretation, given that all other covariates held constant.
- Non-linear term, X^*X , $\log(X)$, etc. can be added.
- Same model diagnostic tools as the linear model (or some variant of them) are also available for logistic regression.



Multiple logistic regression for Default data

Coefficients:

	Estimate	Std. Error	z	value	Pr(> z)
(Intercept)	-1.087e+01	4.923e-01	-22.080	< 2e-16	***
studentYes	-6.468e-01	2.363e-01	-2.738	0.00619	**
balance	5.737e-03	2.319e-04	24.738	< 2e-16	***
income	3.033e-06	8.203e-06	0.370	0.71152	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2920.6 on 9999 degrees of freedom

Residual deviance: 1571.5 on 9996 degrees of freedom

AIC: 1579.5



Classification with logistic regression

- The predicted value of the logistic model can be computed at
 - 1. At linear predictor/logit scale: $\hat{\eta}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$, or
 - 2. At probability scale: $\hat{p}(Y_i = 1 | X_i = x_i) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_i}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_i}}$
- An observation is classified into class $Y_i = 1$ if $\hat{\eta}_i > 0$ or equivalently $\hat{p}(Y_i = 1 | X_i = x_i) > 0.5$
- Other classification threshold can also be applied, based on subject matter knowledge.

Prediction with Default data set

- From the following logistic model

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-5.97524	0.19368	-30.851	< 2e-16 ***
studentYes	-0.64678	0.23625	-2.738	0.00619 **
scale(balance)	2.77483	0.11217	24.738	< 2e-16 ***
scale(income)	0.04046	0.10940	0.370	0.71152

- If someone is a student, balance 1 sd higher than the mean, and income is 1 sd lower from the mean, then that individual's predicted defaults probability is

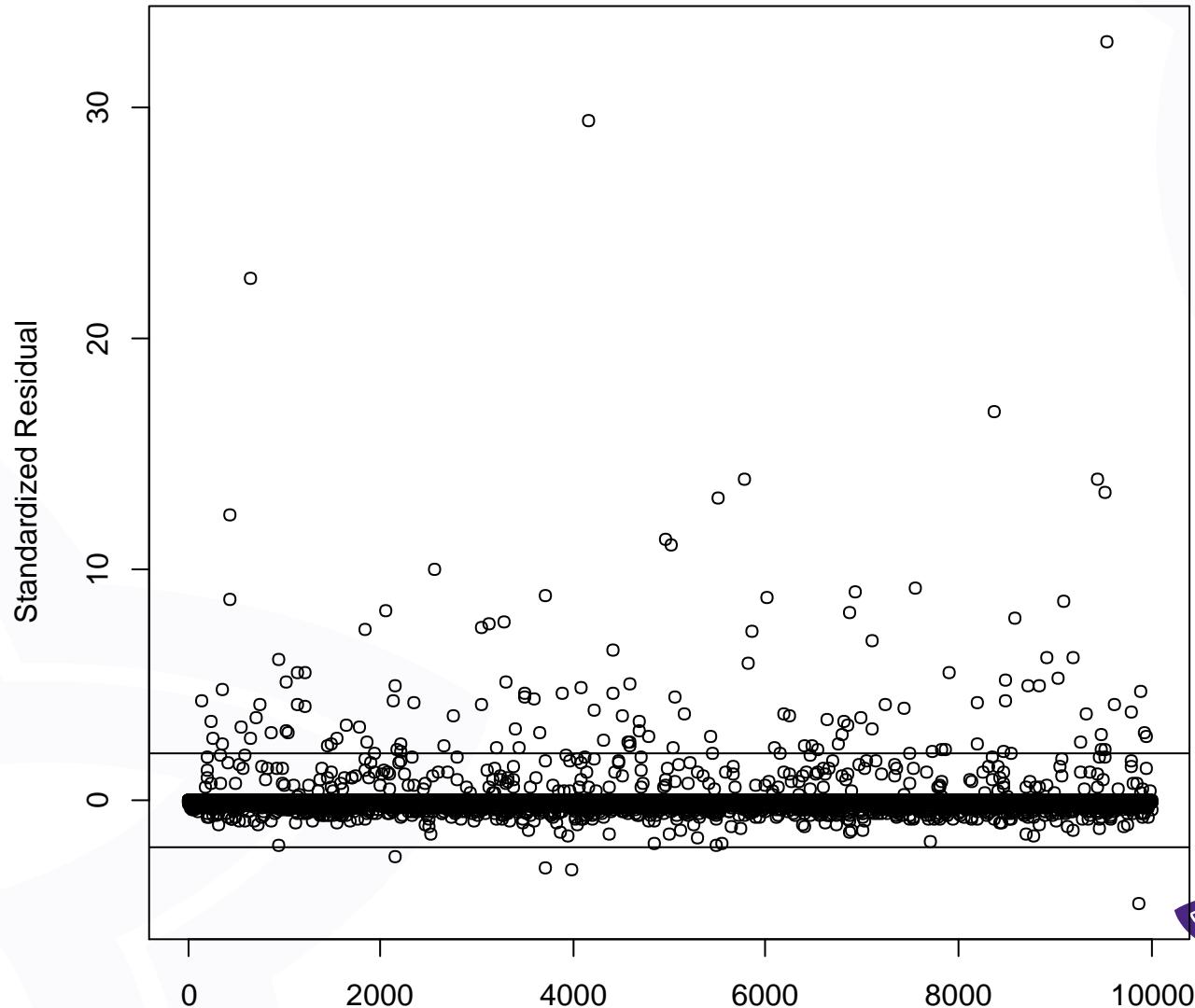
$$p(Y = Yes|X) = \frac{e^{-5.97524 - 0.64678 + 2.77483 + (-1) \times 0.04046}}{1 + e^{-5.97524 - 0.64678 + 2.77483 - 0.04046}}$$

i.e. $p(Y = Yes|X) = 0.02$

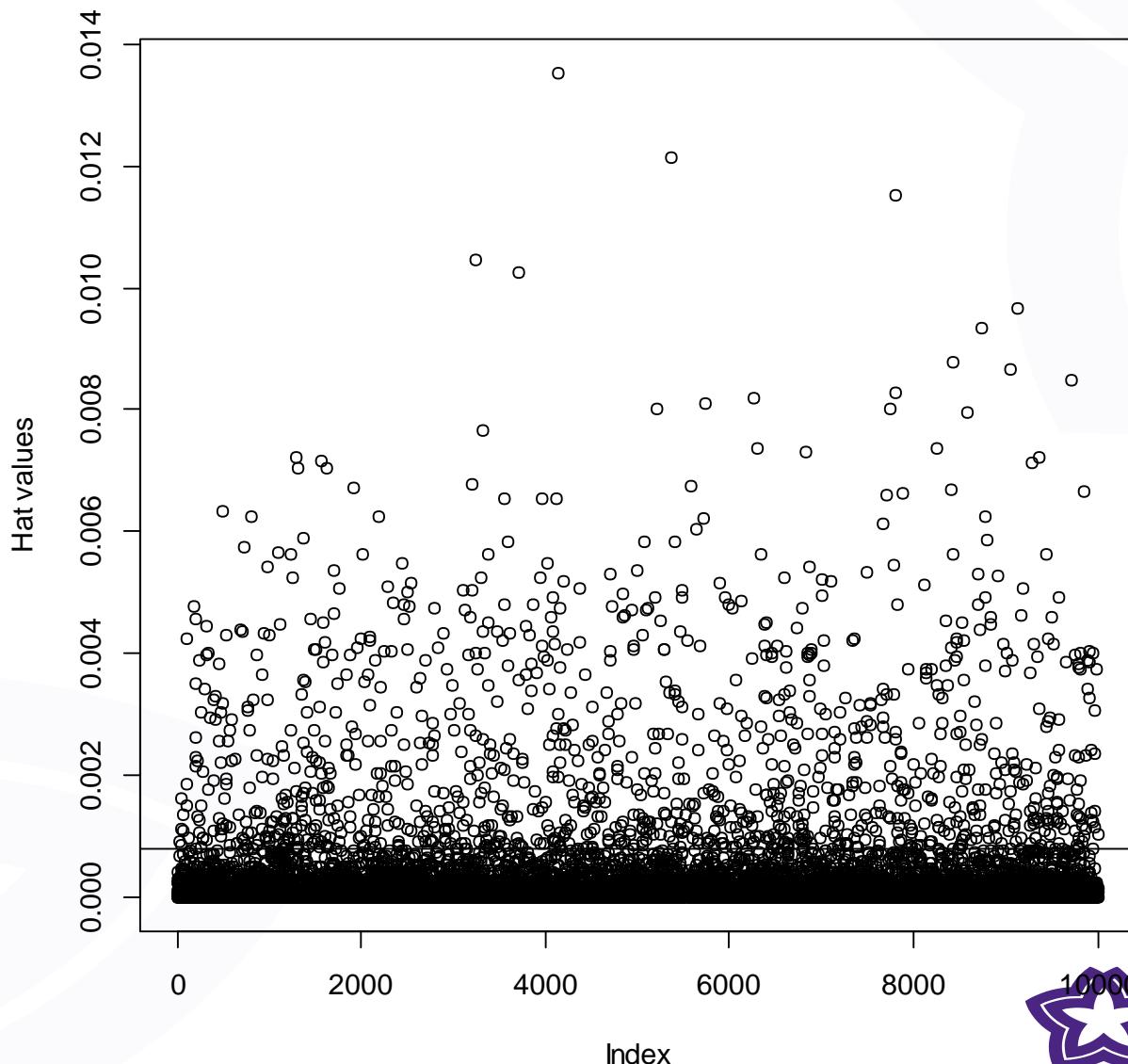
Because, $p(Y = Yes|X) < 0.5$ we classify this individual as non-default.



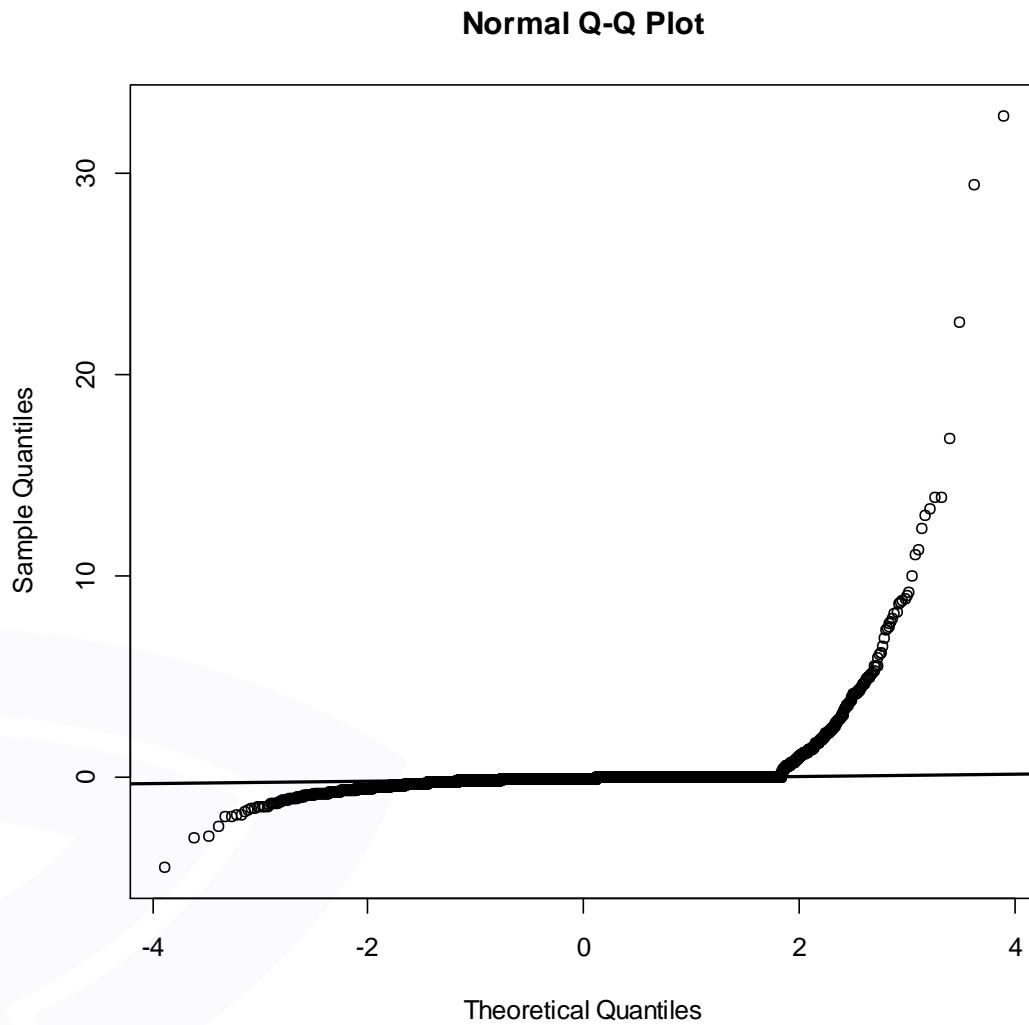
Plot of standardized residuals from the multiple logistic model



Hat-values for logistic regression



Q-Q plot of Pearson's residuals from the logistic model



Classification Beyond Logistic Regression

Statistical Learning (AMI22T)

Lecture: April 14

Moudud Alam (maa@du.se)



HÖGSKOLAN
DALARNA

Contents

- Classification for more than two classes
- Linear discriminant analysis
- Quadratic discriminant analysis
- ROC curve
- K-nearest neighbour classifier
- Cross validation for modes assessment
- Bootstrap



Classification beyond two-class problem

- Assume, a dependent variable contains more than two categories.
- A example could be outcome of a medication: patient's condition improved, not improved, deteriorated or showed severe side effect.
- So far we have seen that logistic regression can deal with only two-class problem.
- There are extensions of logistic regression that can handle more than two class, but they are more suitable for inference than prediction.
- In matter of fact, logistic regression often perform poorly even with two class problem, when classification is the only concern.

Discriminant analysis

- In logistic regression, we solve classification problem using a direct estimate of $p(Y|X)$ from the data.
- In doing so we rely on the assumption that the relationship between logit of $p(Y|X)$ and X is linear.
- In discriminant analysis we do not assume any functional relationship between $p(Y|X)$ and X, but we assume that $f_k(X|Y = k)$ is Gaussian.
- Discriminant analysis attempts to solve classification problem with an indirect estimate of $p(Y|X)$ using Bayes' rule, and relying on direct estimate of parameters in $f_k(X|Y = k)$.

Bayes classifier

- The Bayes' rule is given by

$$p(Y = k|X = x) = \frac{f_k(X = x|Y = k)p(Y = k)}{\sum_k f_k(X = x|Y = k)p(Y = k)}$$

- If we plug-in real data estimate of f_k and $p(Y = k)$ into the above formula, and classify observation as follows

Classify Y_i into class k if $p(Y = k|X = x) > p(Y = m|X = x)$, $\forall k \neq m$

- As such, the above classifier is also called the naïve Bayes classifier.
- For discrete X , computation of Bayes classifier is straightforward but for continuous X it can be problematic because the estimation of f_k is a challenging task.

Linear discriminant analysis

- Assume for simplicity that the response Y is a categorical variable consisting of K classes.
- Assume we have one feature, X .
- Assume the distribution of X in class K is $N(\mu_k, \sigma^2)$.
- Then, using Bayes' rule we have the following likelihood (posterior)

$$L_k = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}{\sum_{k=1}^K \pi_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}$$

- Where π_k is $P(Y = k)$ in the population, called prior (?) probability of class k .

Linear discriminant analysis (cont.)

- Now, we classify an observation, i , to class k if

$$L_{k,i} \geq L_{m,i}, \forall k \neq m; k = 1, \dots, K, m = 1, \dots, K$$

- The above inequality can be presented, after log-transformation, as

$$\log(L_{k,i}) \geq \log(L_{m,i})$$

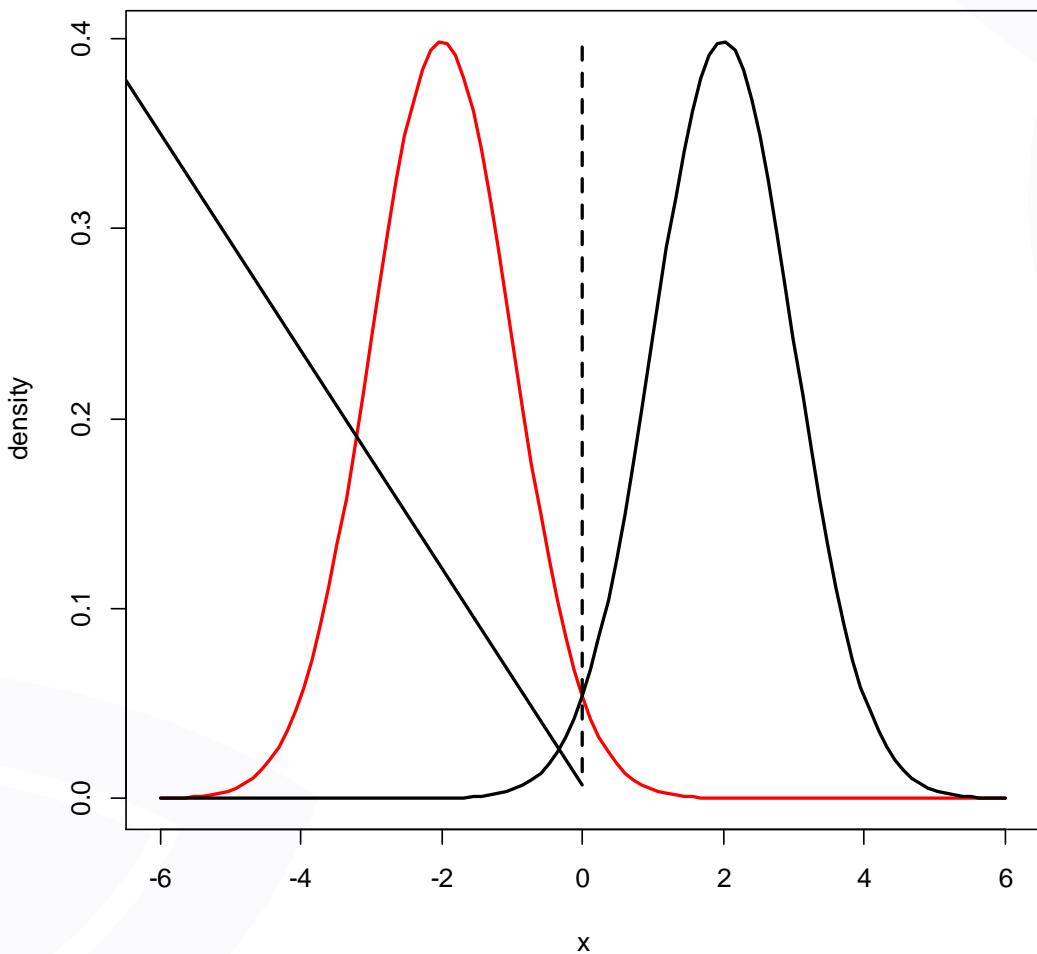
- After some algebra, it turns out that the above classifier assigns an observation to class k if the following is largest

$$\delta_x(k) = x \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

- Notice that the decision rule is a linear function of x .
- For $K = 2$, and $\pi_1 = \pi_2$ the decision boundary is given by $x = \frac{\mu_1 + \mu_2}{2}$.



Visualizing LDA classification



Analysis of Default data set (SLR library)

Call:

```
lda(default ~ ., data = Default)
```

Prior probabilities of groups:

No	Yes
----	-----

0.9667	0.0333
--------	--------

Group means:

student	Yes	balance	income
---------	-----	---------	--------

No	0.2914037	803.9438	33566.17
----	-----------	----------	----------

Yes	0.3813814	1747.8217	32089.15
-----	-----------	-----------	----------

Coefficients of linear discriminants:

LD1

student	Yes	-1.746631e-01
---------	-----	---------------

balance		2.243541e-03
---------	--	--------------

income		3.367310e-06
--------	--	--------------



Training classification error of LDA

```
library(ISLR)
library(MASS)
> data(Default)
> ld1<-lda(default~.,data(Default))
> p2<-predict(ld1,Default)
> table(p2$class,Default$default)

      No  Yes
No  9645 254
Yes   22  79
> prop.table(table(p2$class,Default$default),2)

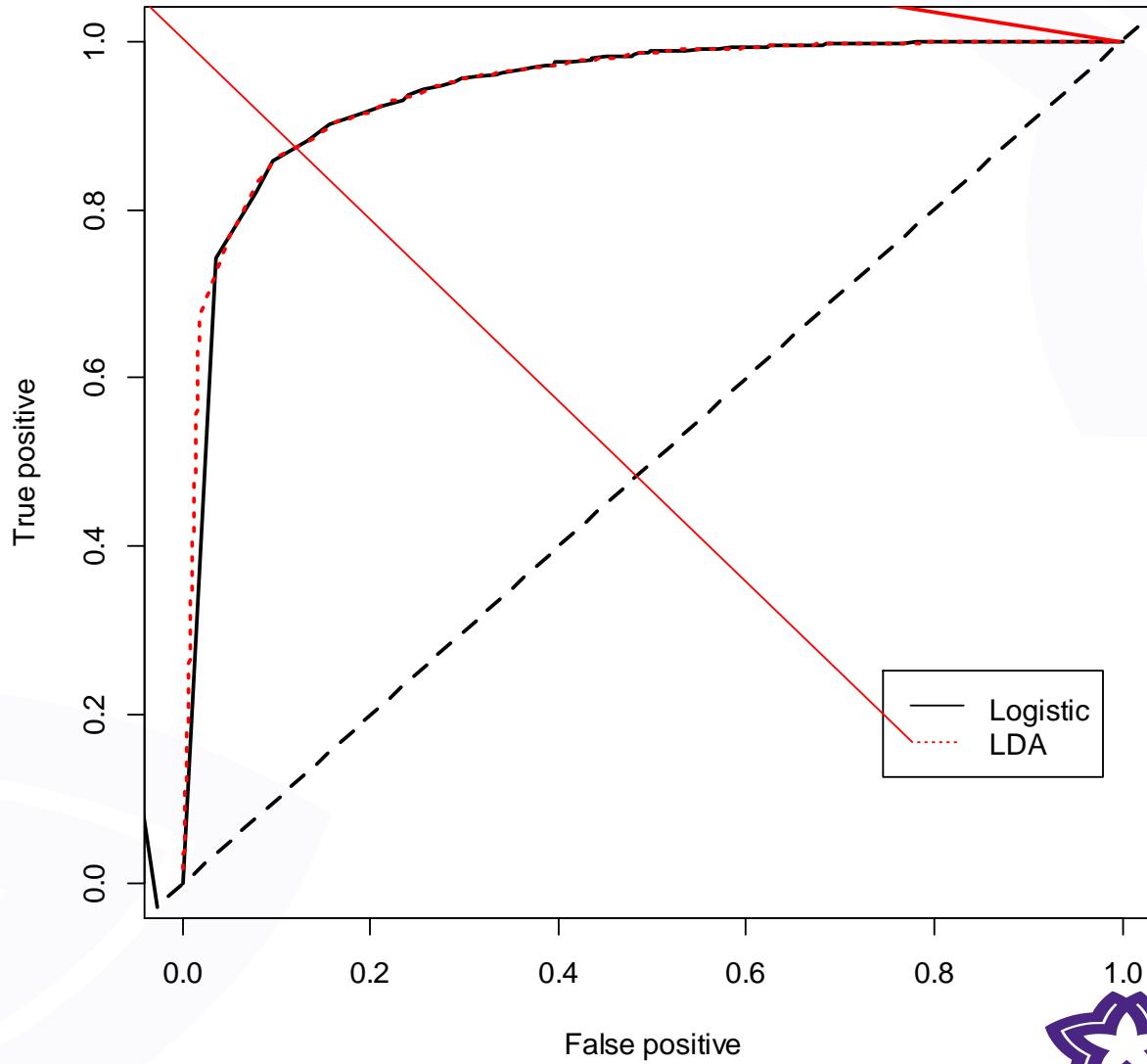
      No      Yes
No 0.997724216 0.762762763
Yes 0.002275784 0.237237237
```

Comparison of LDA with Logistic regression

```
> g1<-glm(default~.,data=Default,family=binomial)
> p1<-ifelse(predict(g1,type="response")>0.5,1,0)
> prop.table(table(p1,Default$default),2)
```

p1	No	Yes
0	0.995862212	0.684684685
1	0.004137788	0.315315315

ROC curves for the logistic regression and LDA



HÖGSKOLAN
DALARNA

Quadratic discriminant analysis

- In LDA we assume equal variance, i.e. $(X|Y = k) \sim N(\mu_k, \sigma^2)$.
- If we drop the equal variance assumption, in LDA, then we obtain quadratic discriminant analysis (QDA).
- Assume we have p input variables $X = (X_1, X_2, \dots, X_p)$, which are distributed as $(X | Y = k) \sim N(\boldsymbol{\mu}_k, \Sigma_k)$ where $\boldsymbol{\mu}_k$ is the mean vector of X and Σ_k is its (variance-) covariance matrix.
- QDA assigns an observation $X_i = \mathbf{x}_i$, to class $Y = k$ if the following discriminant function is maximum for that class

$$\delta_k(\mathbf{x}_i) = (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) - \frac{1}{2} \log |\Sigma_k| + \log(\pi_k)$$



Defaults data example (cont.)

```
> qd1<-qda(default~.,data=Default)
> pre2<-predict(qd1,Default)
> prop.table(xtabs(~pre2$class+Default$default),2)
```

Default\$default

pre2\$class	No	Yes
No	0.996793214	0.717717718
Yes	0.003206786	0.282282282

K-nearest neighbour classification

- K-nearest neighbour (KNN) is a non-parametric classifier, i.e. it does not assume any distribution.
- For an observation i to classify, KNN looks for K other observations, which are nearest among all the observations, with respect to some distance measure.
- Then it classifies observation i to a specific class, based on popularity voting among the k nearest observations.
- Ties are often broken by random assignment.
- Parameter K is subjectively chosen (often via cross validation).

Cross validation

- Models assessment based on Training data can be misleading because of model overfitting.
- It makes better sense if we assess model based on a data that was not used for model estimation (training)
- K-fold cross validation does it in following was
 - Split the data into k subsets of equal size (say n)
 - Use k-1 subsets for the training the remaining one for testing (or prediction)
 - Repeat the above k times such that each of the k subsets is used for training just once.
 - Compute some validation measure, such as mean squared error

$$MSE = \frac{1}{k} \sum_{i=1}^k MSE_i \text{ where } MSE_i = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2$$

- Assess model performance based on the computed measure.
- We can use $k=1$, but $k=10$ is most commonly used

Problem with cross validation

- Results from k-fold cross validation is sensitive to actual splitting of the data, and the choice of k.
- It would be a better choice to study test performance, in sample of size n, with all possible $\binom{n}{n/k}$ test data sets. But, it is often too time consuming.
- What is the optimal split, is also an issue. Typically k=5, and 10 are suggested, on the basis of empirical evidences.
- In some special cases, it is known that k-fold cross validation is upward biased.
- Setting up the best strategy for cross-validation is an active field of research in Statistical and Machine Learning.

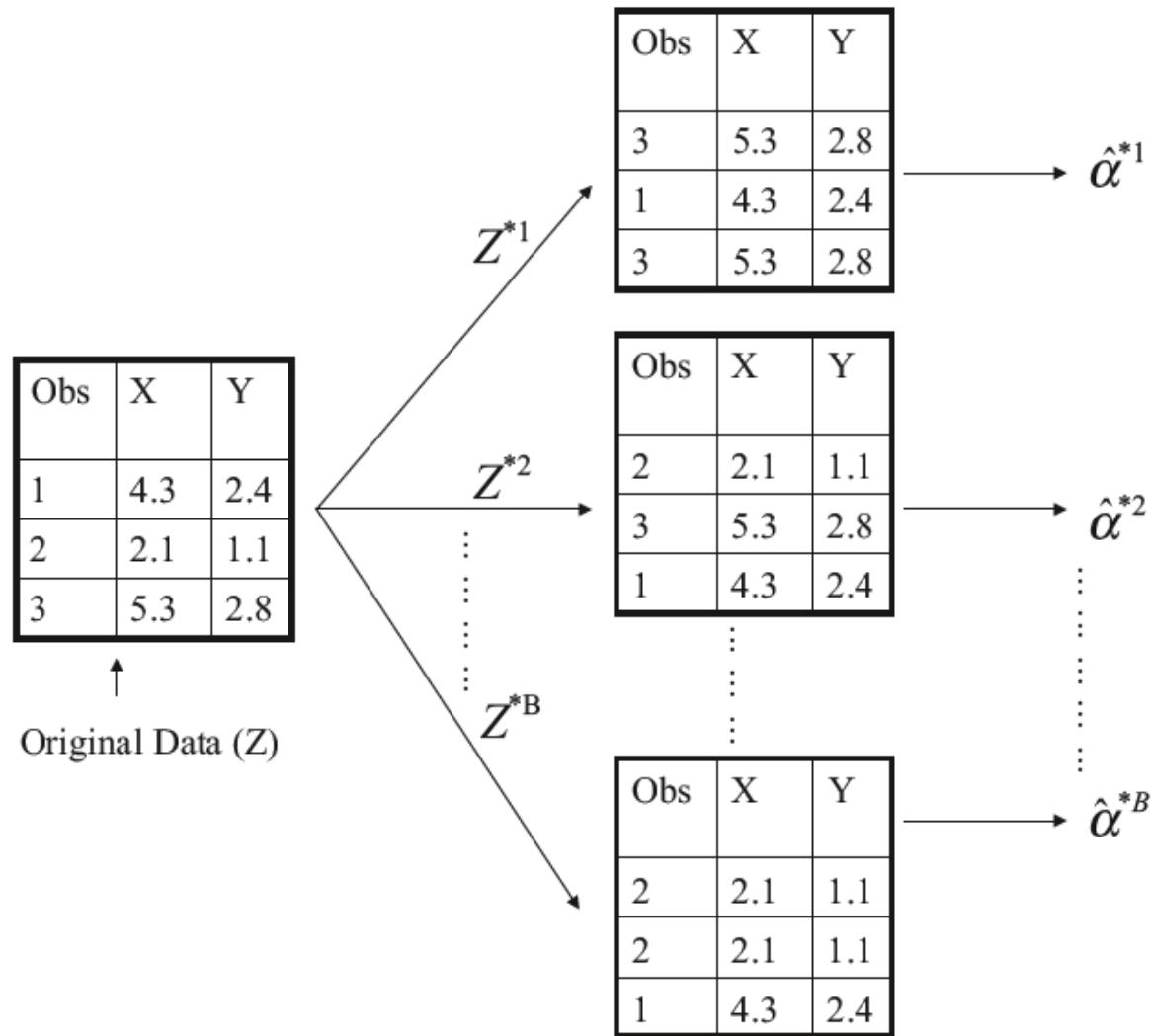


Models assessment vs. uncertainty in estimation

- The process of evaluating model performance is called **model assessment**.
- Process of selecting appropriate level of flexibility of a model is called **model selection**.
- Given a model an estimate of variation of the sample estimates of the model parameters is called uncertainty in estimation.
- In statistical models Standard Error (SE) and confidence interval (CI) are measures of the uncertainty in parameter estimates.
- Instead of relying on theoretical SE and CI, one can repeatedly sample from the original sample, and estimate the same model parameters on all the samples, to study sampling variation.
- The last approach is known as bootstrap.



Bootstrap in picture



Next

- Lab on LDA, QDA, & KNN
- Lab on Cross validation and Bootstrap

Lecture 5 : Model Selection and Regularization

Statistical Learning (AMI22T)

Moudud Alam (maa@du.se)

April 20, 2020



HÖGSKOLAN
DALARNA

Contents

- Shrinkage method
- Feature selection
- Subset selection
- Ridge regression
- Lasso
- Principal component regression



Shrinkage method

- So far, we have seen ordinary least-square (OLS) and maximum likelihood (ML) methods for model parameter estimation.
- These methods do not put any constraint on parameter.
- By constraining parameter, one can achieve higher precision (at the cost some small bias).
- Shrinkage methods seeks model parameter estimates, using OLS or ML, and by shrinking the parameters towards zero.
- Ridge regression and Lasso are two popular shrinkage estimation methods.

Why do we need shrinkage

- If $p > n$ in a linear model, OLS fails to provide unique estimate.
- If some independent variables are highly correlated, OLS (ML) estimator for linear (logistic) regression model becomes numerically unstable.
- If two covariates in a linear (logistic) model are perfectly correlated, the OLS/ML estimation algorithm breaks down.
- OLS and ML method often favour large (and complicated) model, while some of the terms in the models are redundant.
- Removing redundant terms increase interpretability of the model. This is particularly an issue with high dimensional data.

Feature selection

- Feature selection refers to selection of independent variables (or terms), or some linear combination of them so that the model is simplified without compromising (substantially) with its predictive performance.
- There are three approaches for feature selection:
 - Subset selection
 - Regularization (or shrinkage)
 - Dimension reduction
- Certain regularization methods (e.g. Lasso) guarantees faster convergence to the true model, than ordinary ML or OLS.

Subset selection

- Subset selection methods aims to select a subset of the predictors, keeping the statistical model and the estimation method unchanged.
- The methods for subset selection include
 - Best subset selection
 - Forward stepwise selection (or addition)
 - Backward stepwise selection (or deletion)
- R^2 , AIC, BIC etc. are used for comparison of the model fit.

Algorithm 6.1 *Best subset selection*

1. Let \mathcal{M}_0 denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
 2. For $k = 1, 2, \dots, p$:
 - (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
 - (b) Pick the best among these $\binom{p}{k}$ models, and call it \mathcal{M}_k . Here *best* is defined as having the smallest RSS, or equivalently largest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Algorithm 6.2 *Forward stepwise selection*

1. Let \mathcal{M}_0 denote the *null* model, which contains no predictors.
 2. For $k = 0, \dots, p - 1$:
 - (a) Consider all $p - k$ models that augment the predictors in \mathcal{M}_k with one additional predictor.
 - (b) Choose the *best* among these $p - k$ models, and call it \mathcal{M}_{k+1} . Here *best* is defined as having smallest RSS or highest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Algorithm 6.3 Backward stepwise selection

1. Let \mathcal{M}_p denote the *full* model, which contains all p predictors.
 2. For $k = p, p-1, \dots, 1$:
 - (a) Consider all k models that contain all but one of the predictors in \mathcal{M}_k , for a total of $k-1$ predictors.
 - (b) Choose the *best* among these k models, and call it \mathcal{M}_{k-1} . Here *best* is defined as having smallest RSS or highest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Model selection criteria

- Indirect criteria based on training data only
 - Adjusted R²: $1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}$
 - AIC: $-2 \times \log(Likelihood) + 2 \times \text{numer of parameter(s)}$
 - Cp: $\frac{1}{n}(RSS + 2p\hat{\sigma}^2)$
 - BIC: $-2 \times \log(Likelihood) + \log(n) \times \text{numer of parameter(s)}$
- Direct estimate of test error
 - Validation set approach
 - Cross validation
 - Often, one-sd rule is used for comparing test errors

Shrinkage method 1: Ridge regression

- Recall with $RSS = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{1,i} - \cdots - \beta_p x_{p,i})^2$, OLS estimates β_j 's ($j = 0, 1, \dots, p$) parameters by minimizing RSS.
- Ridge regression estimates parameters by minimizing

$$RSS + \lambda \sum_{j=1}^p \beta_j^2$$

- The parameter $\lambda \geq 0$ is called tuning parameter, and the term $\|\beta\| = \sqrt{\lambda \sum_{j=1}^p \beta_j^2}$ is called penalty factor.
- With $\lambda = 0$ ridge estimator is OLS but when $\lambda \rightarrow \infty$ ridge estimator approaches to 0.
- Before running a ridge regression, it is customary to scale all the independent variables (why?).



Shrinkage method 2: Lasso

- Ridge regression shrinks the parameter but not draw them to exactly 0.
- This can be problematic, when we have a lot of independent variables but a few of them are to be related to the response.
- Lasso solves this problem by sharply pulling the small estimates to zero.
- In doing so, Lasso estimates parameters by minimizing

$$RSS + \lambda \sum_{j=1}^p |\beta_j|$$

- The term $\lambda \sum_{j=1}^p |\beta_j|$ is called L_1 penalty.

Comparison between OLS, Ridge, and Lasso

- Assume the following model: $y_i = \beta_i + \epsilon_i$
- OLS estimate of β_i is obtained by minimizing $\sum_{i=1}^n (y_i - \beta_i)^2$ and gives $\hat{\beta}_i = y_i$, which is a perfect fit with the training data.
- Ridge regression estimates parameter by minimizing $\sum_{i=1}^n (y_i - \beta_i)^2 + \lambda \sum_{i=1}^n \beta_i^2$, and gives $\hat{\beta}_i^R = \frac{y_i}{1+\lambda}$
- Lasso estimates parameter by minimizing $\sum_{i=1}^n (y_i - \beta_i)^2 + \lambda \sum_{i=1}^n |\beta_i|$, and gives

$$\hat{\beta}_i^R = \begin{cases} y_i - \frac{\lambda}{2}; & \text{if } y_i > \lambda/2 \\ y_i + \frac{\lambda}{2}; & \text{if } y_i < -\lambda/2 \\ 0; & \text{if } |y_i| \leq \lambda/2 \end{cases}$$



Analysis of credits data set

Effects	Ridge	Lasso	OLS
(Intercept)	0	0	-479.208
Income	0.208	0	-7.803
Limit	0.005	0	0.191
Rating	0.085	0	1.137
Cards	6.267	0	17.724
Age	0.702	0	-0.614
Education	5.170	15.667	-1.099
GenderFemale	8.101	0	-10.653
GenderMale	7.114	0NA	
StudentYes	8.980	0	425.747
MarriedYes	9.595	0	-8.5339
EthnicityAsian	4.990	0	16.804
EthnicityCaucasian	7.48533	0	10.107
Lambda	34001.89	1062.803	0



Summarizing shrinkage method

- Shrinkage method can be used for feature selection.
- Shrinkage estimation has a constrained minimization explanation.
- Shrinkage methods have a Bayesian interpretation (we skip this).
- Tuning parameters in the shrinkage method are selected using cross validation method.
- An R package, **glmnet**, has functions **glmnet** and **cv.glmnet** which can be used for shrinkage estimation.
- These methods are not limited to linear models.
- Besides ridge regression and Lasso, there are other shrinkage methods too, e.g. adaptive Lasso, elastic net, SCAD, hierarchical likelihood method (we skip these).



Dimension reduction

- Assume we have p independent variable X_1, X_2, \dots, X_p .
- Dimension reduction methods works in the following way.
- First, it tries to find out some linear combinations of X's

$$Z_j = \sum_{k=1}^p \phi_{j,k} x_k, \quad j=1, 2, \dots, M; \quad M < p$$

- Then in the second step, in stead of fitting p variable regression model, fit a smaller M variable model

$$y_i = \theta_0 + \sum_{j=1}^M \theta_j z_j$$

- Optimal $\phi_{j,k}$ is computed via Principal Component Analysis (PCA), leading to principal component regression (PCR) and partial least square (PLS) methods.



Principal Component Analysis

- PCA is an unsupervised learning technique that aims to reduce the dimensionality of an $n \times p$ matrix, X.
- The first principal component Z_1 is chosen so that highest variation of the data lies in the direction of Z_1 .
- With $Z_j = \sum_{k=1}^p \phi_{j,k}(X_k - \bar{X}_k)$, $\phi_{j,k}$ are called loadings which are simply the components of the j:th eigen-vector of $Var(X)$.
- Properties of eigen-vector assures that Z_j 's are orthogonal to each other, removing any collinearity.
- M components are chosen such that they jointly explain almost all the variation in X.

Example of PCA

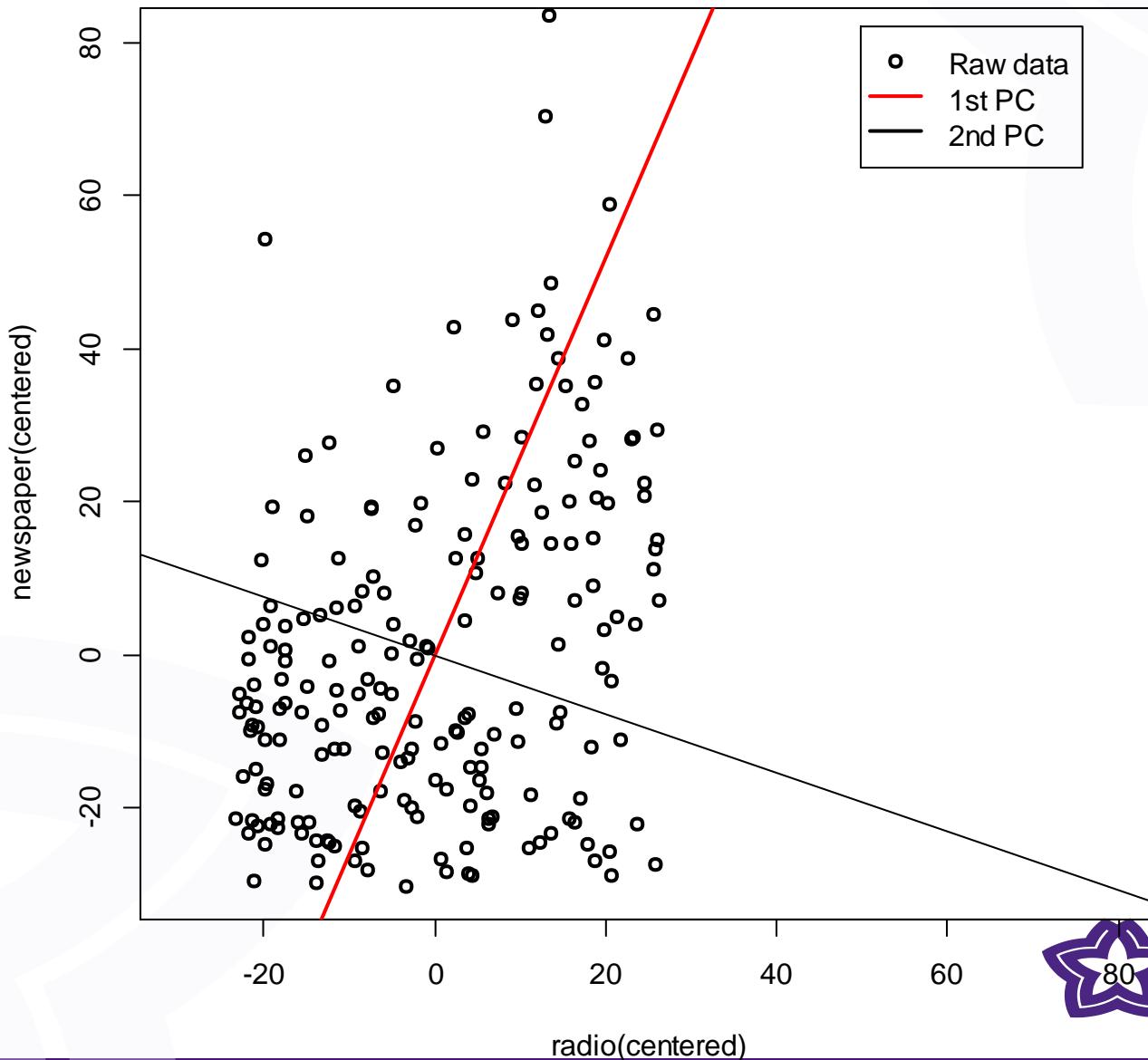
- Consider the **radio** and **newspaper** variables in advertising data set that we analysed in Lecture 1.
- Correlation between these two variables is 0.35.
- Covariance matrix is given by

	radio	newspaper
radio	220.4277	114.4970
newspaper	114.4970	474.3083

- Eigen vectors of the (co-) variance matrix are

	[,1]	[,2]
[1,]	0.3587726	-0.9334250
[2,]	0.9334250	0.3587726

Visualizing PCs of radio and newspaper variables



Example of PCR: Advertising data (ISLR website)

```
> p1<-pcr(sales~.,data=Adds,ncomp=3)
```

```
> summary(p1)
```

Data: X dimension: 200 4

Y dimension: 200 1

Fit method: svdpc

Number of components considered: 3

TRAINING: % variance explained

	1 comps	2 comps	3 comps
X	64.60	94.08	98.46
sales	61.28	62.00	69.57



Lecture 6: Non-linear models

Statistical Learning (AMI22T)

April 24, 2020

Moudud Alam (maa@du.se)



HÖGSKOLAN
DALARNA

Contents

- Non-linear models
- Orthogonal polynomial
- Step function
- Basis and natural spline
- Smoothing spline
- Generalized additive models (GAM)
- R examples using Wage data



From linear to non-linear relationship

- We have already seen how to use, higher order terms of the independent variables, e.g. X^2 , X^3 , etc., to model non-linear relationship.
- Here, we will see more parsimonious representations of non-linear relationship such as
 - Orthogonal polynomial
 - Piece-wise constant, and polynomial models
 - Regression splines
 - Smoothing splines
 - Local regression

Orthogonal Polynomial

- We have seen polynomial regression: $y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$
- Assume "X" is randomly and independently drawn from $N(0,1)$.
- Still there is a big chance that the terms X, X², and X³ can be highly correlated which is not desirable in linear regression e.g.

```
set.seed(220)
x<-rnorm(100)
> cor(x,x^3)
[1] 0.8669517
```

- Orthogonal polynomial of degree 3 will fit the following model

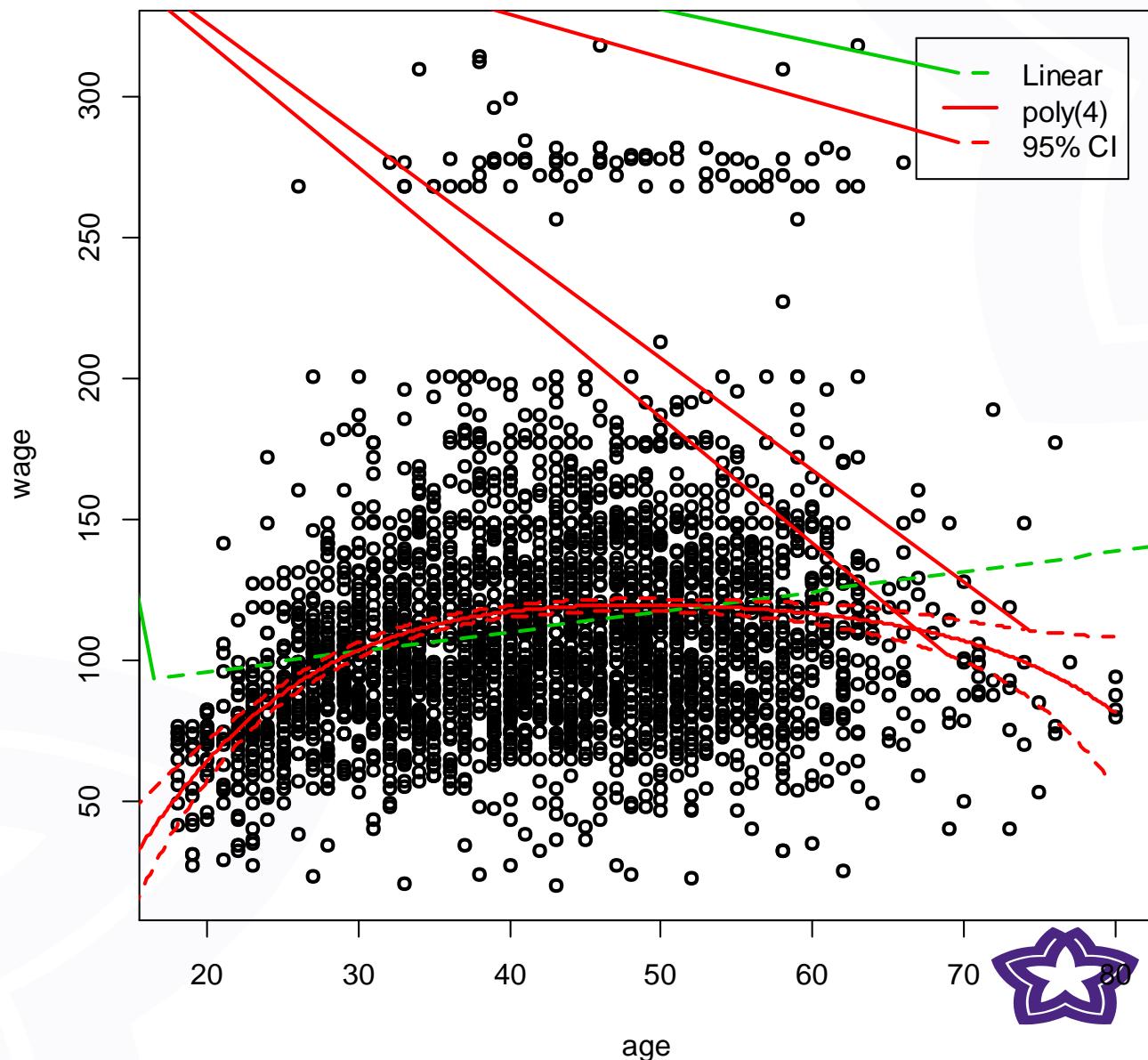
$$y = \beta_0 + \beta_1 P_1(x) + \beta_2 P_2(x, x^2) + \beta_3 P_3(x, x^2, x^3) + \epsilon$$

where the P_1 , P_2 , and P_3 , are orthogonal (uncorrelated) to each other.

- An R function poly(), does this job.



Linear vs. polynomial regression with Wage data set



Piece wise constant (step function) regression

- A piece wise constant break the range of X into bins and different constant for different bin.
- Assume we break X at k break points (c_1, c_2, \dots, c_k) as

$$\begin{aligned}C_0(X) &= I(X < c_1), \\C_1(X) &= I(c_1 \leq X < c_2), \\C_2(X) &= I(c_2 \leq X < c_3), \\&\vdots \\C_{K-1}(X) &= I(c_{K-1} \leq X < c_K), \\C_K(X) &= I(c_K \leq X),\end{aligned}$$

- The above equation can be written mathematically as

$$y_i = \beta_0 + \sum_{j=1}^K \beta_j C_j(x_i) + \epsilon_i \quad (1)$$

Moving beyond step function

- A general form of Model (1) can be written as

$$y_i = \beta_0 + \sum_{j=1}^K \beta_j b_j(x_i) + \epsilon_i \quad (2)$$

where b_j is called basis function

- Denoting $b_j = c_j = I(c_{j-1} \leq x_i < c_j)$ we get back to Model (1). The points, c_1, c_2, \dots, c_K are called knots.
- If we take $b_j(x_i) = x_i^j$ we get polynomial regression.
- With $K = 1$, if we take $b_1 = \beta_{1,0} + \beta_{1,1}x_i + \beta_{1,2}x_i^2$, when $x_i < c_1$, and $b_2 = \beta_{2,0} + \beta_{2,1}x_i + \beta_{2,2}x_i^2$, with $\beta_0 = 0$, and $\beta_j = 1$ (for parameter identification), then we get two piece quadratic model.
- In (1), and piecewise polynomial (2), regression lines are not continuous but we might want a continuous regression curve.



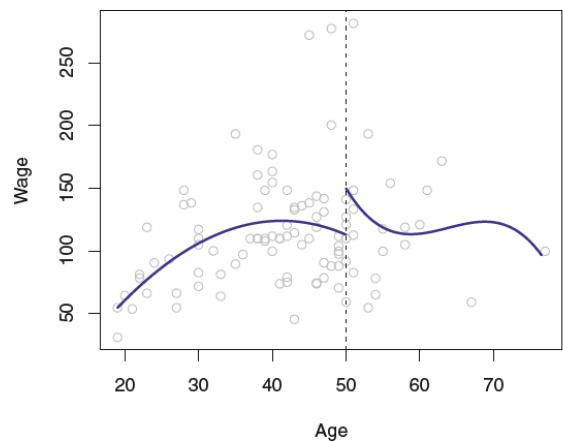
Spline

- A spline function $b_s(x)$ of degree n with knots at $\tau_i, i = 1, 2, \dots, m; a = \tau_0 < \tau_1 \dots < \tau_{m+1} = b$ has the following properties
 - $b_s(x)$ is a polynomial of degree (power) not exceeding n on each sub-interval $[\tau_{i-1}, \tau_i]; 1 \leq i \leq m + 1$
 - $b_s(x)$ has continuous derivatives up to order $n-1$, on $[a, b]$
- The above properties imply that
 - A spline function is a piecewise polynomial
 - The various polynomial segments (of degree n) are joined together at knots.
 - The entire function is (smoothly) continuous, including at the knots.
- A spline with $n=3$, is called cubic spline, and widely used for regression type of models.
- Natural spline is a spline function that puts additional linearity constraints at the end points.

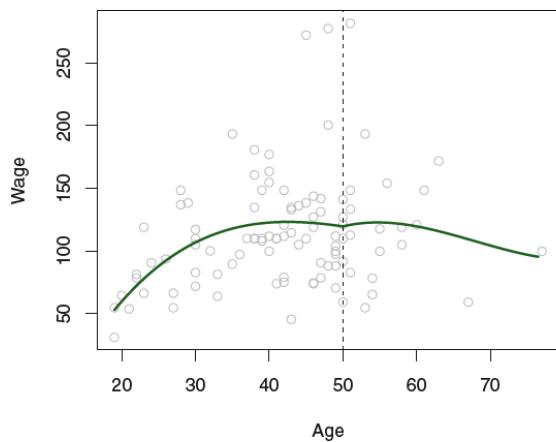


Examples of splines and other piecewise regressions

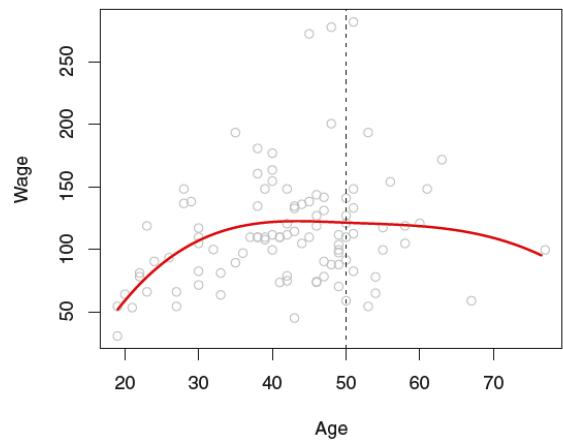
Piecewise Cubic



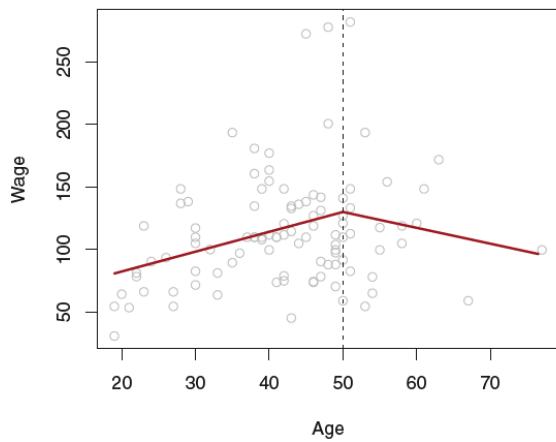
Continuous Piecewise Cubic



Cubic Spline



Linear Spline



Parameters in a cubic spline

- There are many ways to represent a cubic spline.
- The most common way to represent a cubic spline with knots at $\xi_i; i = 1, \dots, K$ is as follows

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \sum_{j=1}^K \beta_{j+3} (x_i - \xi_k)_+^3$$

where $(x - a)_+ = x - a$ if $x > a$ and zero otherwise.

- The above representation implies that we need to estimate $4+K$ parameters.
- The number of (effective) parameters is referred to as the (effective) degrees of freedom (df).
- Because splines often include many parameters, regularization is helpful in parameter estimation.



Choosing the knots

- For spline (and other piecewise regression) models, the selection of knots is a crucial issue.
- In principle, one should set the knots at the points where the relationship changes its trend.
- One can look at bivariate plot of predictor versus response, and set more knots around the locations where the relationship changes quickly and fewer knots where the relationship is stable.
- However, the above is very subjective.
- A popular practise it to set the knots uniformly over the distribution of X, e.g. at 25%, 50% and 75% percentile.
- Notice that there will be two additional knots, at the both ends of X.

Degrees of freedom for splines

- A cubic basis spline with K knots have $(K+2)+4$ parameters, which is called its degrees of freedom (df). $K=3$ gives 9 df.
- However, the R function `bs()` does not count the intercept as a parameter, it also ignores the two boundary knots, ending up with 6 df. Note: It is an internal construction of `bs` function.
- For natural cubic spline, there will be two additional constraints for the boundary knots, requiring (for $K=3$) 4 degrees of freedom in `ns()` function.
- Notice that the actual (effective) degrees of freedom is different from what we specify in `df` argument in the respective R functions.

Smoothing spline

- Smoothing spline attempts to fit a model $y_i = g(x_i) + \epsilon_i$.
- For this the OLS objective (loss) function is $\sum_{i=1}^n (y_i - g(x_i))^2$.
- However, minimizing the above loss function would give a perfect fit with the training data.
- Therefore, smoothing spline minimize the following objective functions

$$\sum_{i=1}^n (y_i - g(x_i))^2 + \lambda \int (g''(t))^2 dt$$

where λ is a tuning parameter.

- In other word, smoothing spline fits a regularized version of the cubic natural spline with a knot at each data point.



Choosing tuning parameter in smoothing spline

- In smoothing spline we do not have to select knots, but we need to select the tuning parameter λ .
- Notice that $\lambda = 0$ gives no smooth, and $\lambda = \infty$ gives a straight line.
- Even though $n=K$, all the parameters are heavily shrunken (depending on λ). Therefore, it is not fair to say that these models have 0 residual df.
- Instead, we use the concept of effective degrees of freedom. Let

$$\hat{\mathbf{g}} = S_\lambda \mathbf{y}$$

Where, S_λ is a function of X and λ producing predicted value at some scale. For OLS regression $S_{\lambda=0} = X(X^T X)^{-1} X^T$ giving $\hat{\mathbf{y}} = S_\lambda \mathbf{y}$.

- Define effective df as: $df_\lambda = \sum_{i=1}^n (S_\lambda)_{i,i}$.
- We can choose df_λ using cross validation, e.g. LOOCV.



Generalize Additive Model (GAM)

- GAM is a framework for fitting a class of non-linear statistical model, e.g.

$$y_i = \beta_0 + \sum_j f_j(X_{j,i}) + \epsilon_i$$

where f_j is some non-linear function e.g. natural spline, basis spline, etc.

- It is called additive because it takes one f_j for one feature X_j .
- Because of additivity, we can easily examine the effect of one variable, keeping other fixed.
- In R, `gam()` function in package `gam` can fit GAM models.

Example: Analysing Wage data set

- We analyse wage with an orthogonal polynomial of degree 4

```
Summary(lm(formula = wage ~ poly(age, 4), data = Wage))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	111.7036	0.7287	153.283	< 2e-16 ***
poly(age, 4)1	447.0679	39.9148	11.201	< 2e-16 ***
poly(age, 4)2	-478.3158	39.9148	-11.983	< 2e-16 ***
poly(age, 4)3	125.5217	39.9148	3.145	0.00168 **
poly(age, 4)4	-77.9112	39.9148	-1.952	0.05104 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 39.91 on 2995 degrees of freedom

Multiple R-squared: 0.08626, Adjusted R-squared: 0.08504

F-statistic: 70.69 on 4 and 2995 DF, p-value: < 2.2e-16

> AIC(l1)

[1] 30641.11

Selecting degree of polynomial

```
data(Wage)
l1<-lm(wage~poly(age,4),data=Wage)
l3<-lm(wage~poly(age,3),data=Wage)
l5<-lm(wage~poly(age,5),data=Wage)
> anova(l3,l1,l5)
Analysis of Variance Table
Model 1: wage ~ poly(age, 3)
Model 2: wage ~ poly(age, 4)
Model 3: wage ~ poly(age, 5)
  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1  2996 4777674
2  2995 4771604  1   6070.2 3.8098 0.05105 .
3  2994 4770322  1   1282.6 0.8050 0.36968
---

```

So, we go for degree 4

Fitting spline with Wage data

```
bs1<-lm(wage~bs(age,knots=c(25,40,60)),data=Wage)
```

```
summary(bs1)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	60.494	9.460	6.394	1.86e-10 ***
bs(age, knots = c(25, 40, 60))1	3.980	12.538	0.317	0.750899
bs(age, knots = c(25, 40, 60))2	44.631	9.626	4.636	3.70e-06 ***
bs(age, knots = c(25, 40, 60))3	62.839	10.755	5.843	5.69e-09 ***
bs(age, knots = c(25, 40, 60))4	55.991	10.706	5.230	1.81e-07 ***
bs(age, knots = c(25, 40, 60))5	50.688	14.402	3.520	0.000439 ***
bs(age, knots = c(25, 40, 60))6	16.606	19.126	0.868	0.385338

Residual standard error: 39.92 on 2993 degrees of freedom

Multiple R-squared: 0.08642, Adjusted R-squared: 0.08459

F-statistic: 47.19 on 6 and 2993 DF, p-value: < 2.2e-16

```
> AIC(bs1)
```

```
[1] 30644.59
```

Fitting natural spline (with 3 equidistance knots)

```
ns1<-lm(wage~ns(age,df=4),data=Wage)
```

```
summary(ns1)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	58.556	4.235	13.827	<2e-16 ***
ns(age, df = 4)1	60.462	4.190	14.430	<2e-16 ***
ns(age, df = 4)2	41.963	4.372	9.597	<2e-16 ***
ns(age, df = 4)3	97.020	10.386	9.341	<2e-16 ***
ns(age, df = 4)4	9.773	8.657	1.129	0.259

Residual standard error: 39.92 on 2995 degrees of freedom

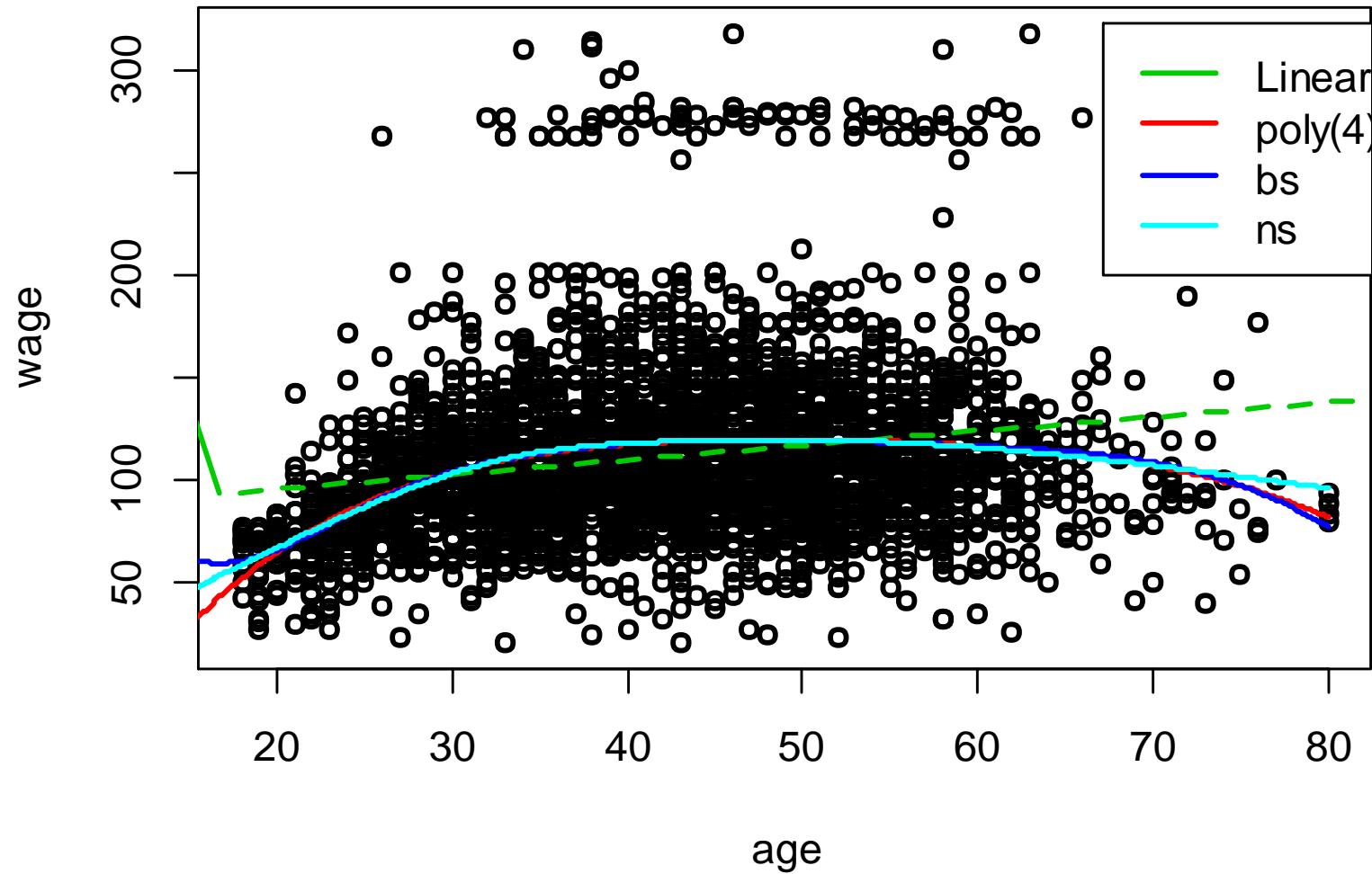
Multiple R-squared: 0.08598, Adjusted R-squared: 0.08476

F-statistic: 70.43 on 4 and 2995 DF, p-value: < 2.2e-16

```
> AIC(ns1)
```

```
[1] 30642.05
```

Comparison of fits with training data



Fitting a Gam model with smoothing spline for age, and education as factor

```
g1<-gam(wage~s(age,5)+education,data=Wage)
```

```
Summary(g1)
```

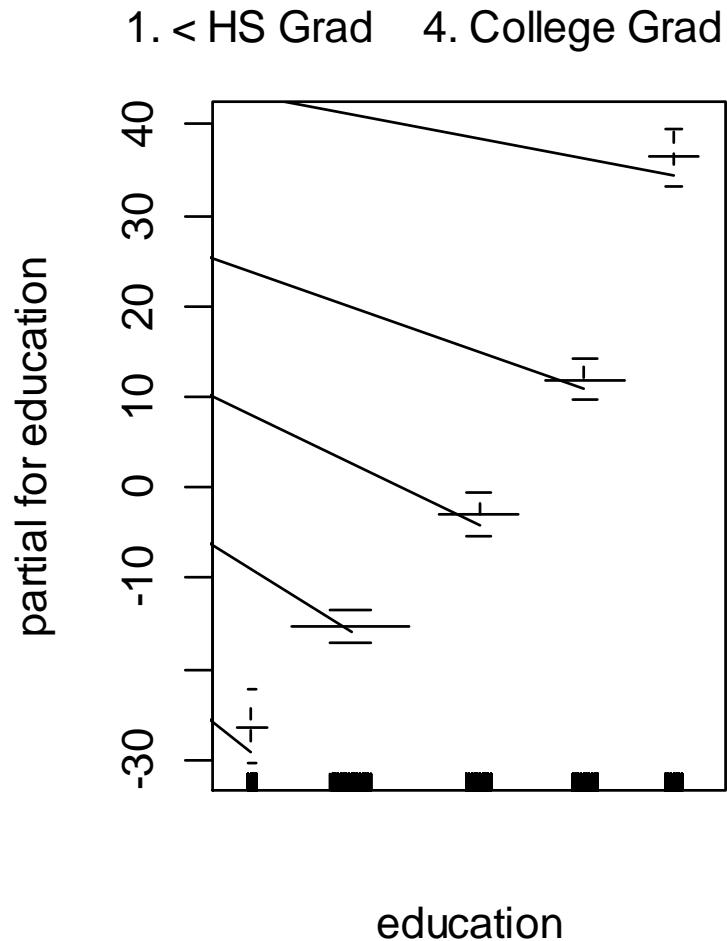
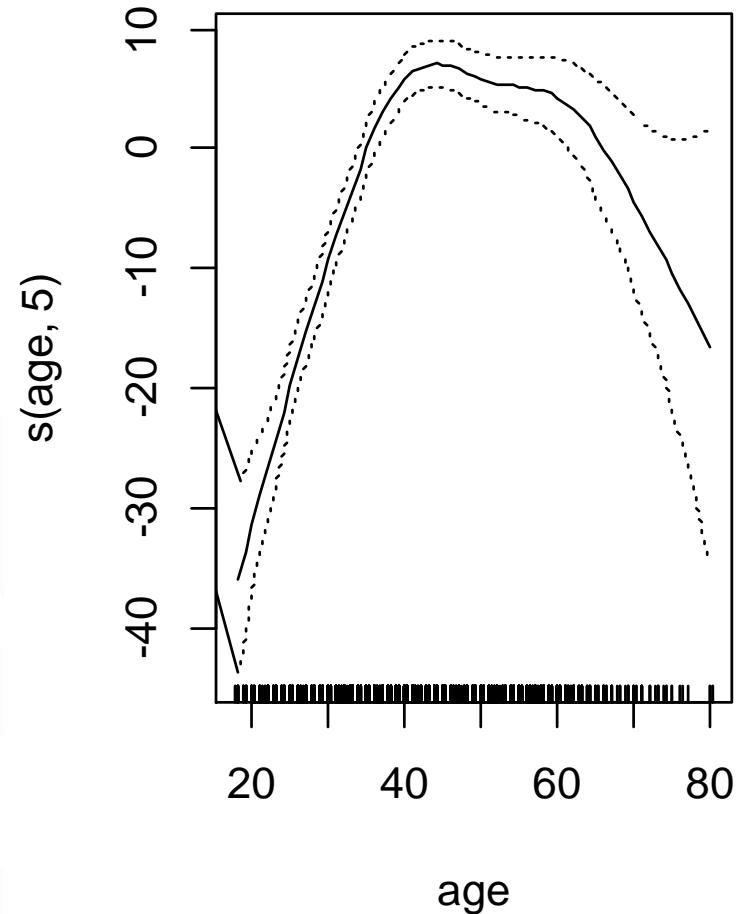
Anova for Parametric Effects

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
s(age, 5)	1	199870	199870	161.01	< 2.2e-16 ***
education	4	1073789	268447	216.25	< 2.2e-16 ***
Residuals	2990	3711731	1241		

Anova for Nonparametric Effects

	Npar	Df	Npar F	Pr(F)
(Intercept)				
s(age, 5)		4	31.472	< 2.2e-16 ***
education				

Effects plot of the GAM model



Decision trees

Ilias Thomas

Outline

- Regression trees
- Classification trees

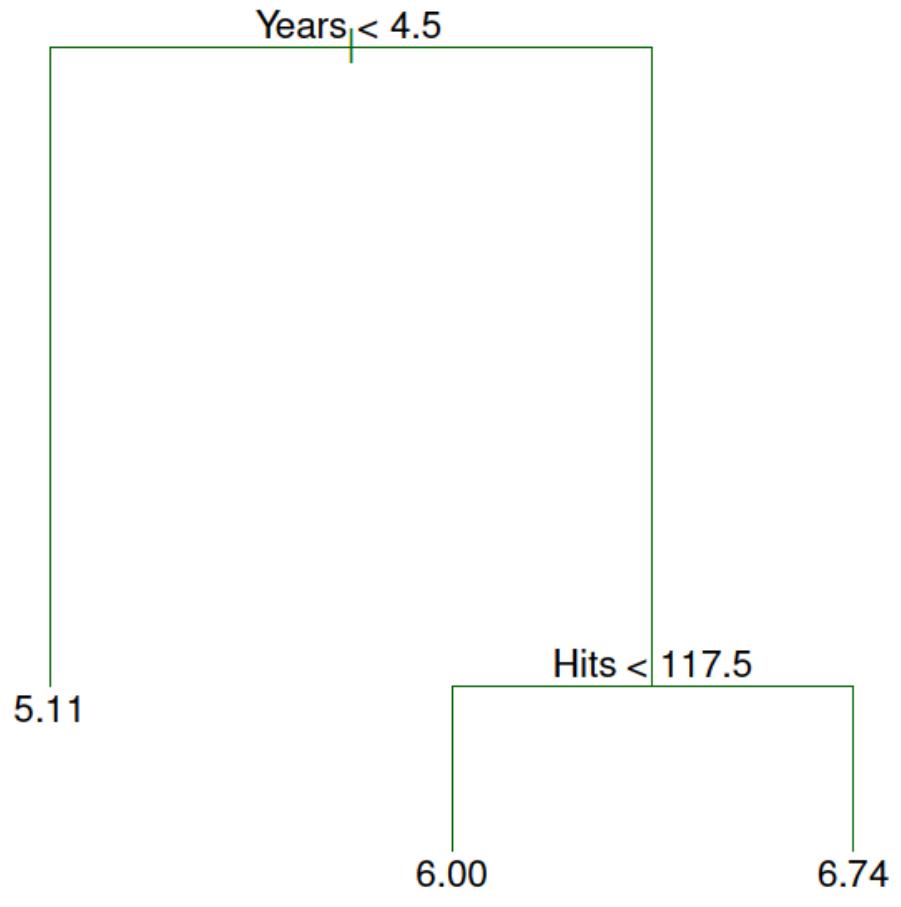
Introduction

- Segment the predictor space into a number of simple regions or boxes.
- Use the mean of the predictor in the box as final prediction (if numerical).
- Use a majority vote of the category of the box (if categorical).

Basic notions

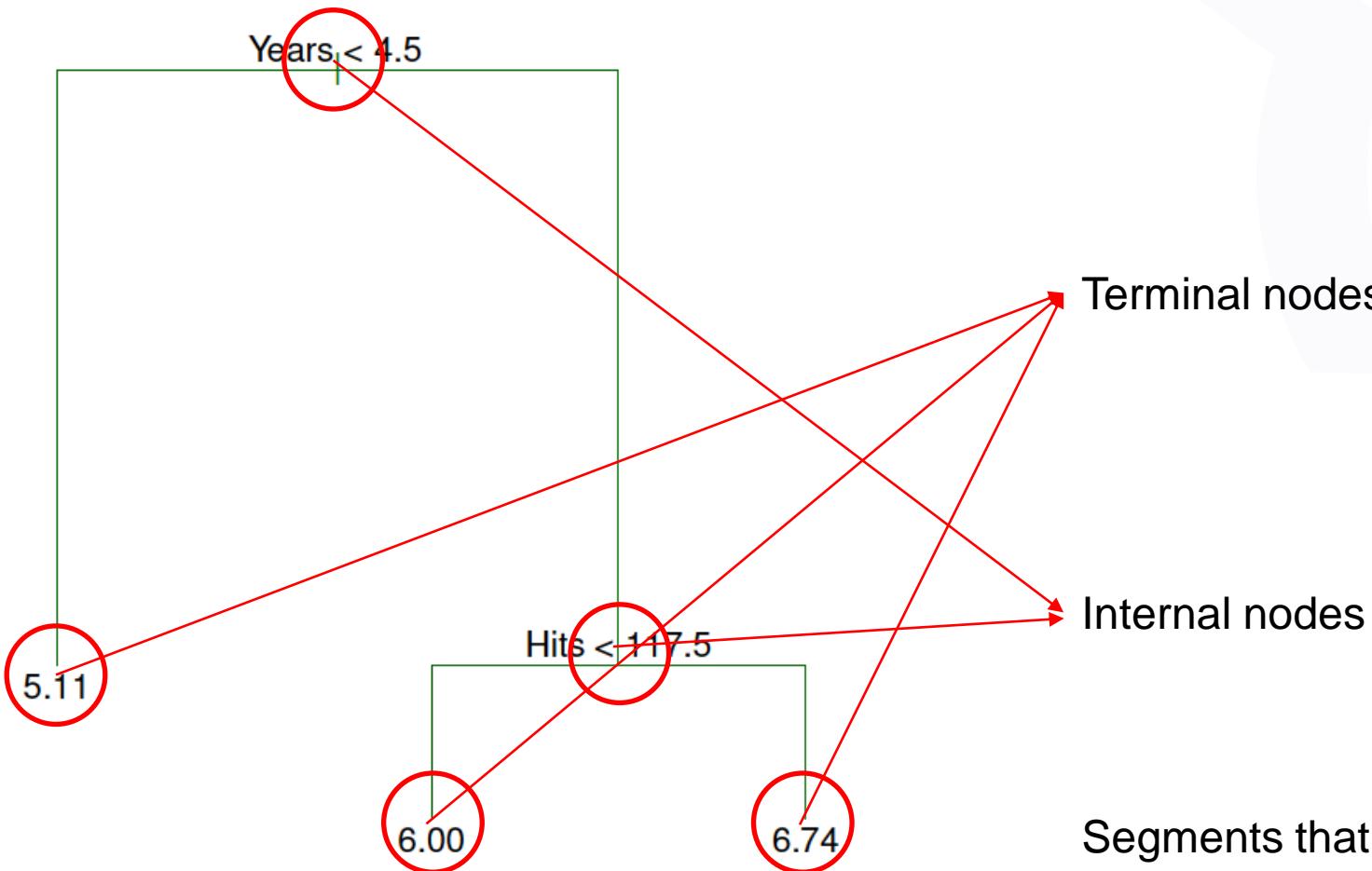
- A series of splitting rules.
- These is always a binary split.
- Based on the splits we can draw a tree.

Basic notions



- This is the Hitters dataset.
- There are rules to predict the salary of a player.
- The rules are based on years on the league and hits the player had the previous year.

Basic notions

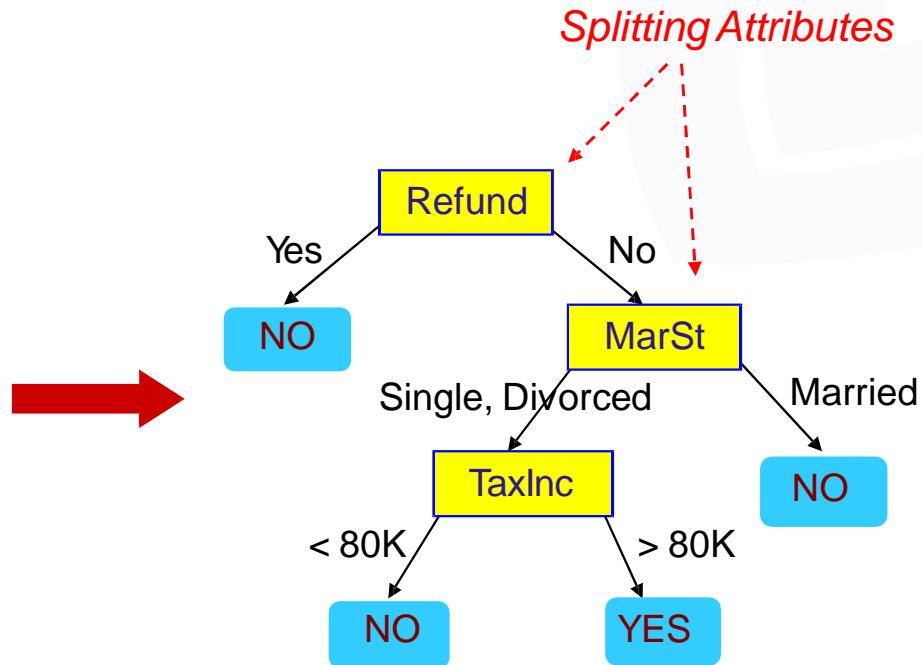


Segments that connect
the nodes are called
branches (green on the
figure)

Example of a Decision Tree

					categorical	categorical	continuous	class
Tid	Refund	Marita l Status	Taxable Income	Cheat				
1	Yes	Single	125K	No				
2	No	Married	100K	No				
3	No	Single	70K	No				
4	Yes	Married	120K	No				
5	No	Divorced	95K	Yes				
6	No	Married	60K	No				
7	Yes	Divorced	220K	No				
8	No	Single	85K	Yes				
9	No	Married	75K	No				
10	No	Single	90K	Yes				

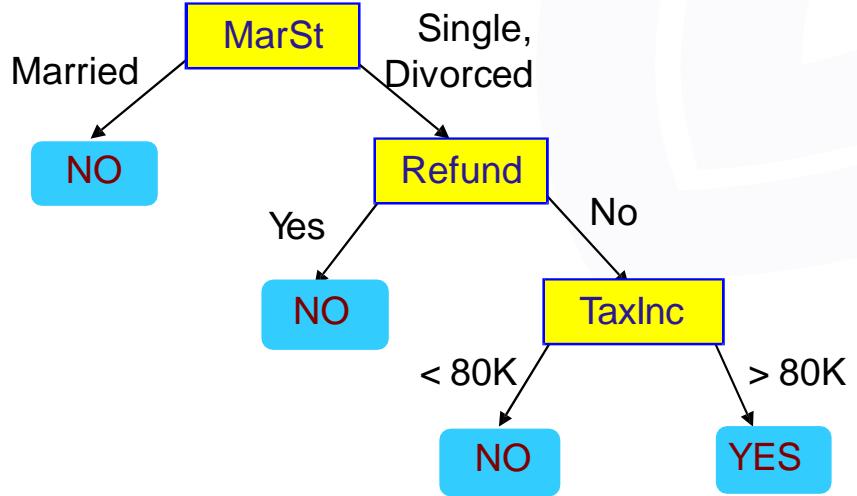
Training Data



Model: Decision Tree

Another Example of a Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat	categorical	categorical	continuous	class
1	Yes	Single	125K	No				
2	No	Married	100K	No				
3	No	Single	70K	No				
4	Yes	Married	120K	No				
5	No	Divorced	95K	Yes				
6	No	Married	60K	No				
7	Yes	Divorced	220K	No				
8	No	Single	85K	Yes				
9	No	Married	75K	No				
10	No	Single	90K	Yes				



There could be more than one tree that fits the same data!

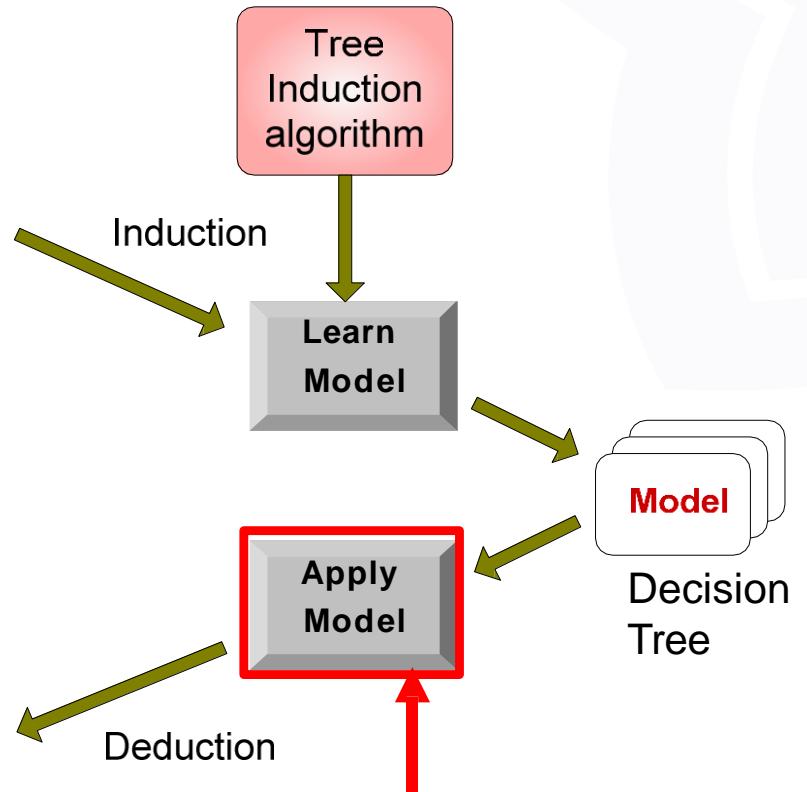
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

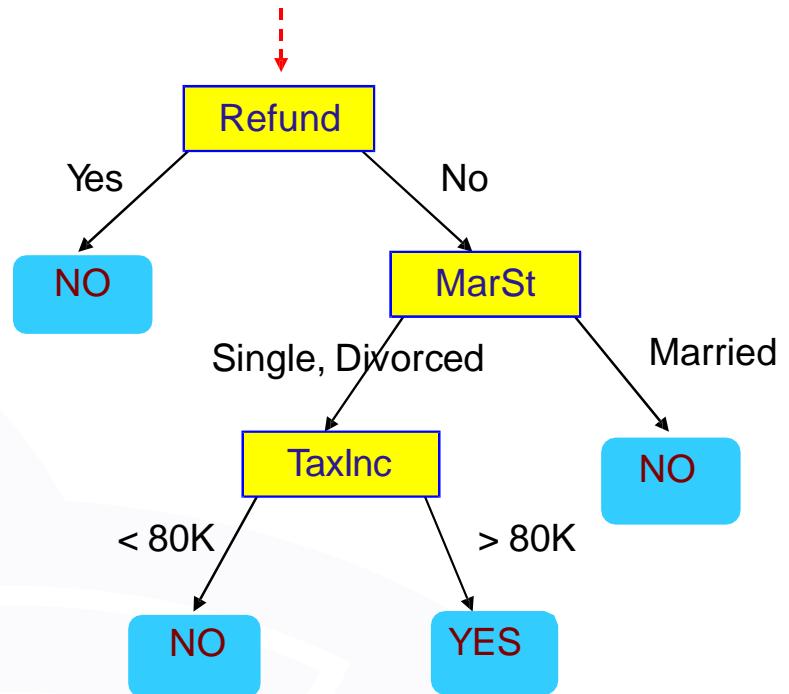
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Apply Model to Test Data

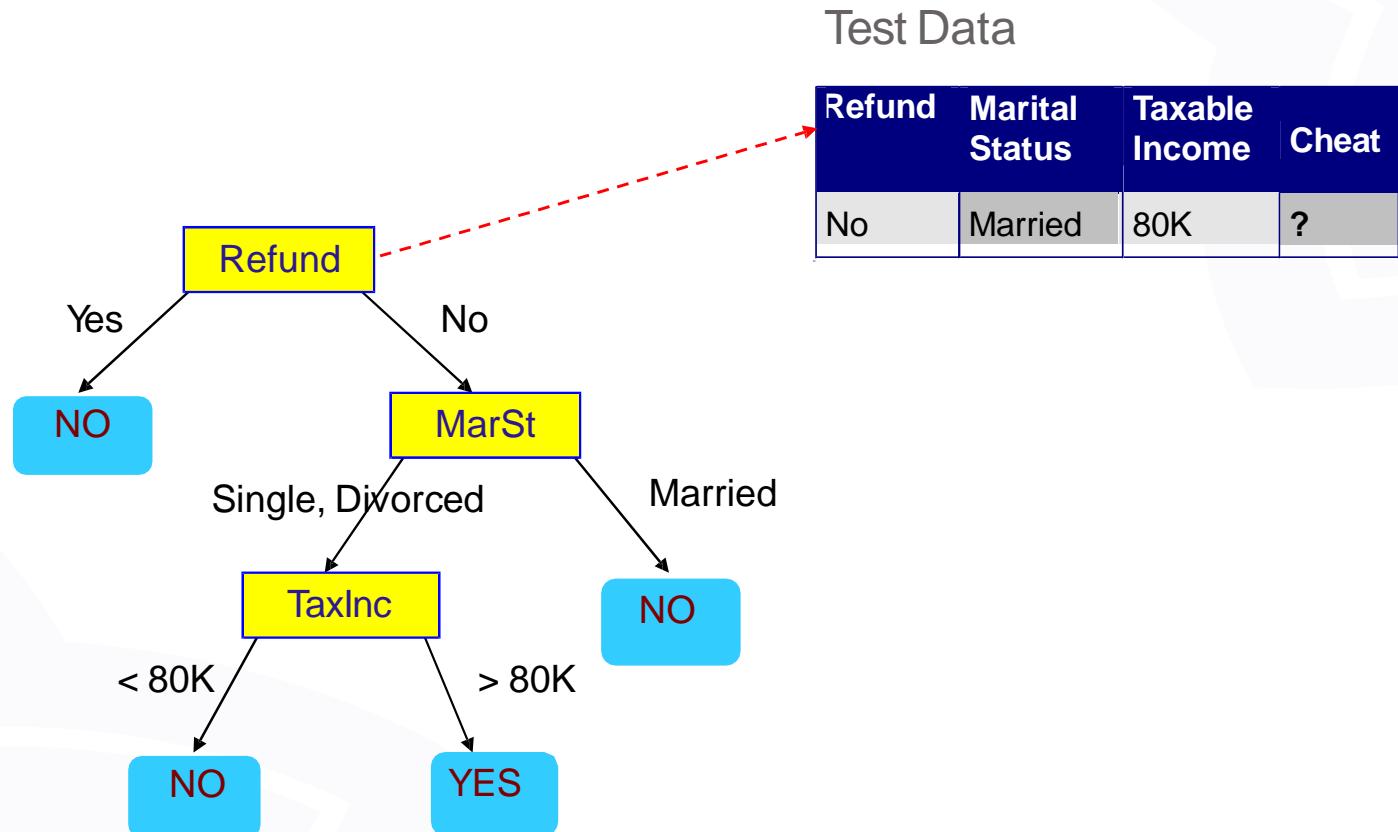
Start from the root of tree.



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

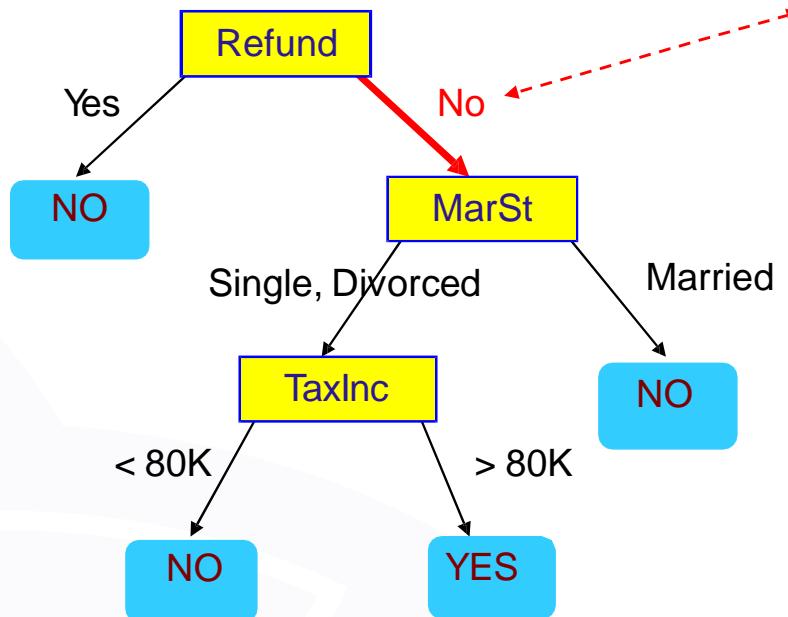
Apply Model to Test Data



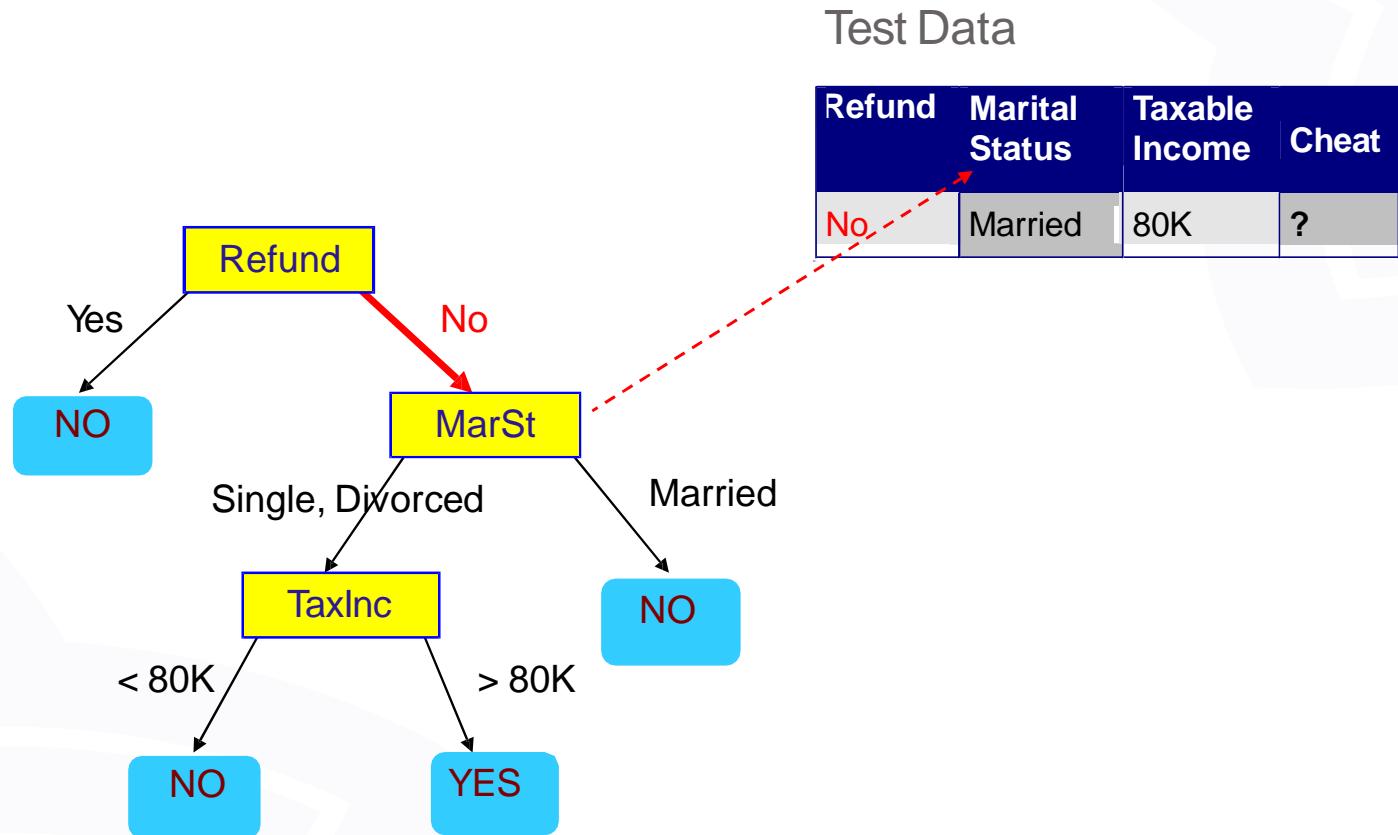
Apply Model to Test Data

Test Data

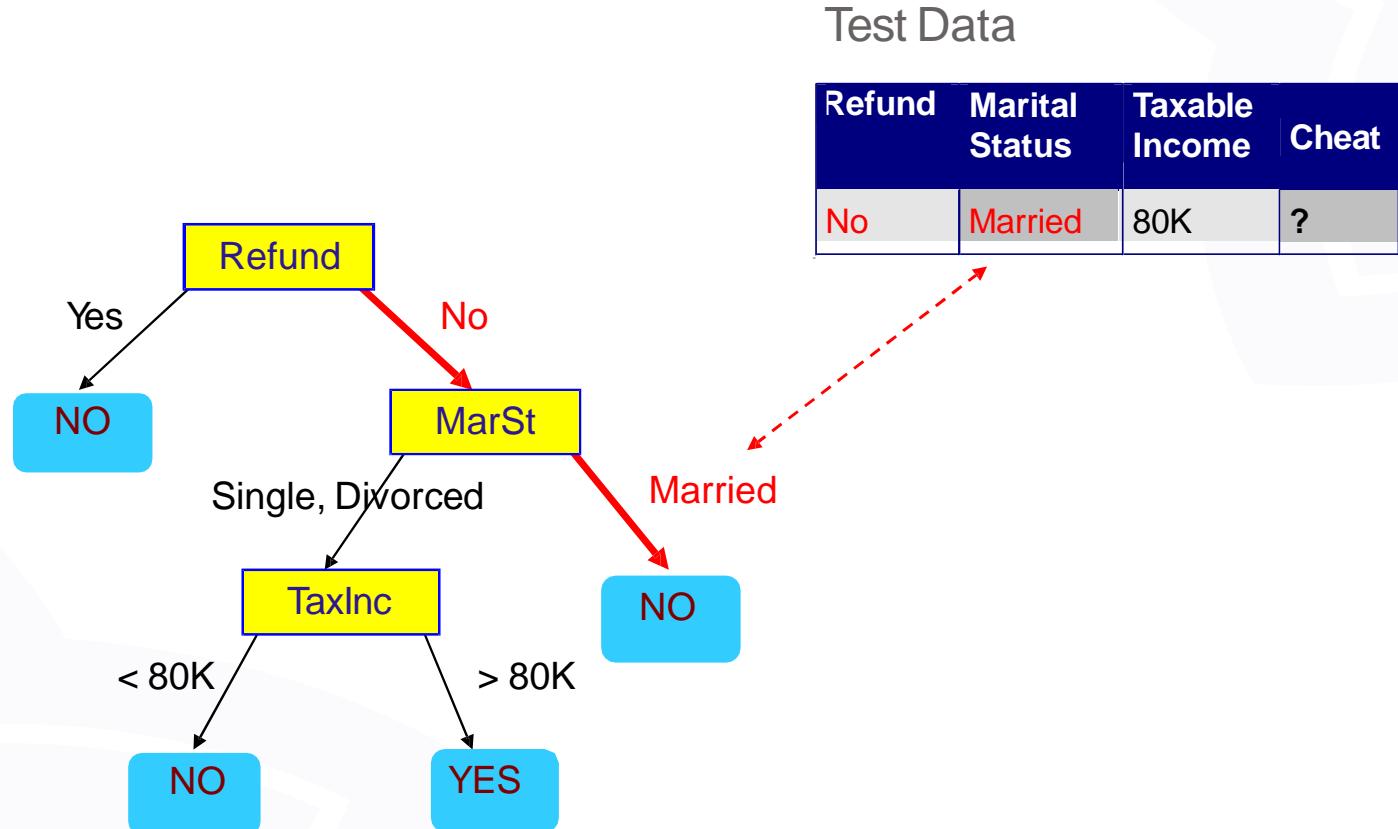
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



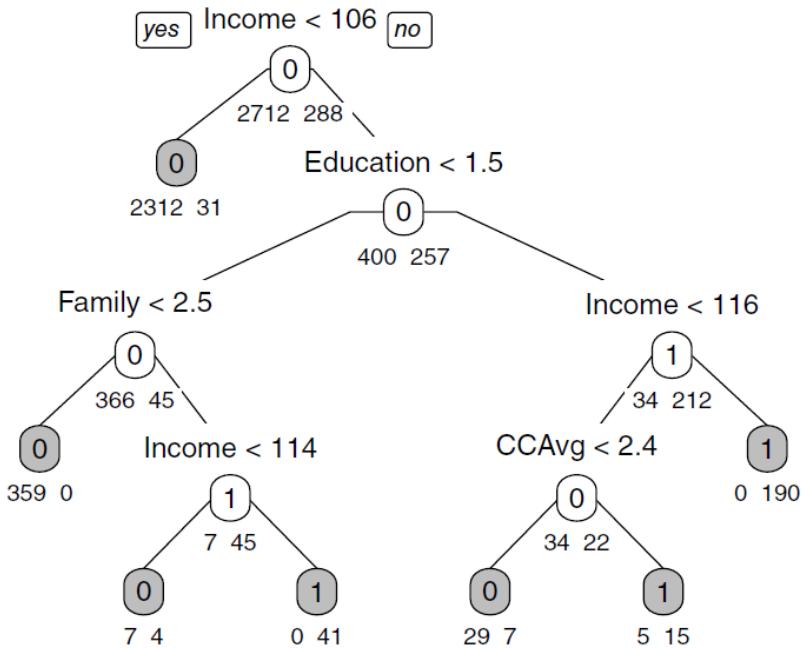
Apply Model to Test Data



Apply Model to Test Data



Example – Tree into rules



IF($Income \geq 106$) AND ($Education < 1.5$) AND ($Family < 2.5$)
THEN $Class = 0$ (nonacceptor).

Key Ideas

Recursive partitioning: Repeatedly split the records into two parts so as to achieve maximum homogeneity within the new parts

Pruning: Simplify the tree by pruning peripheral branches to avoid overfitting

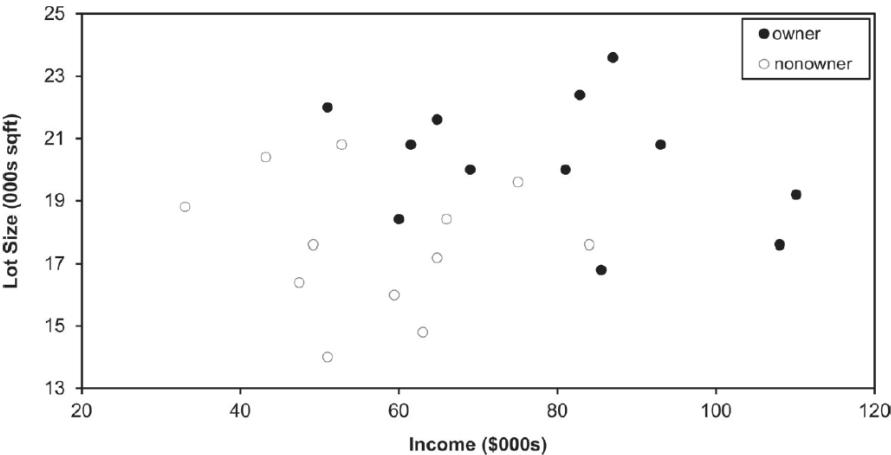
Recursive Partitioning

Example: Riding Mowers

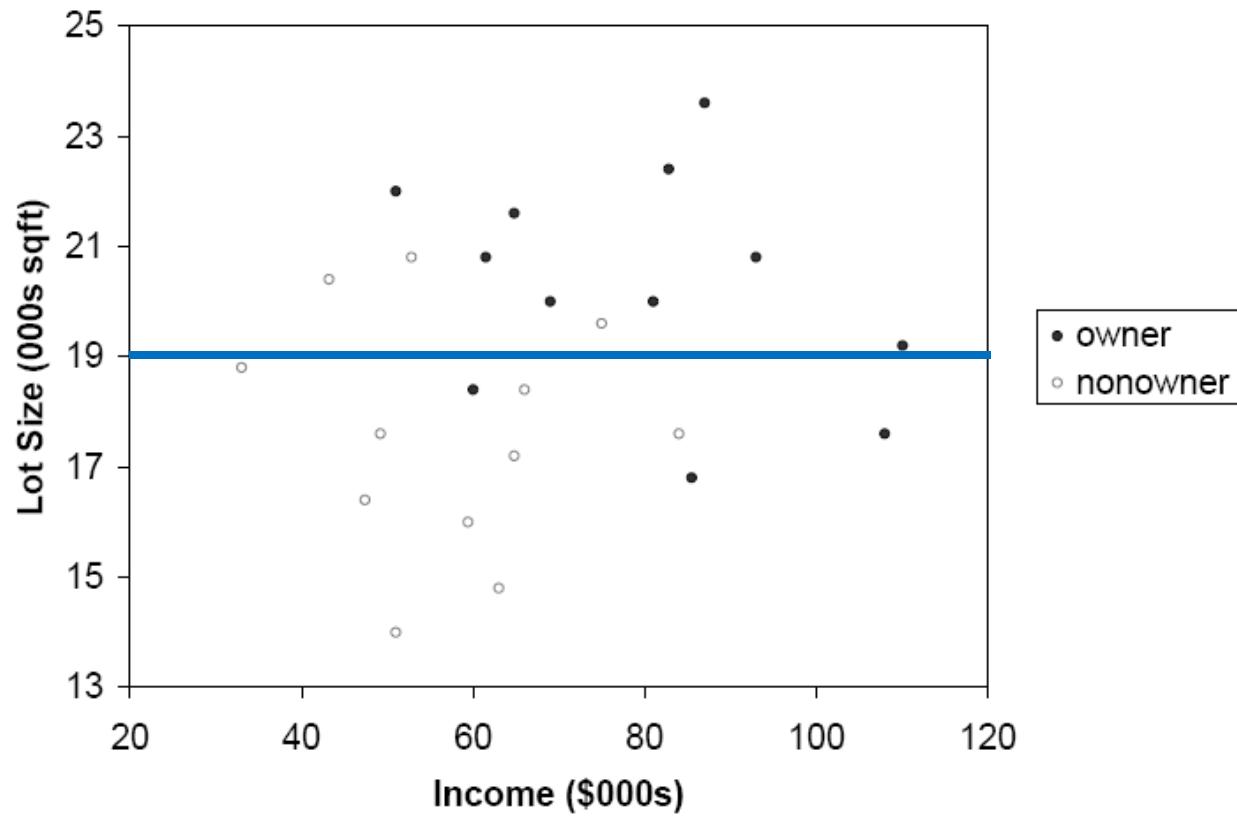
- Goal: Classify 24 households as owning or not owning riding mowers
- Predictors = Income, Lot Size

Example: Riding Mowers

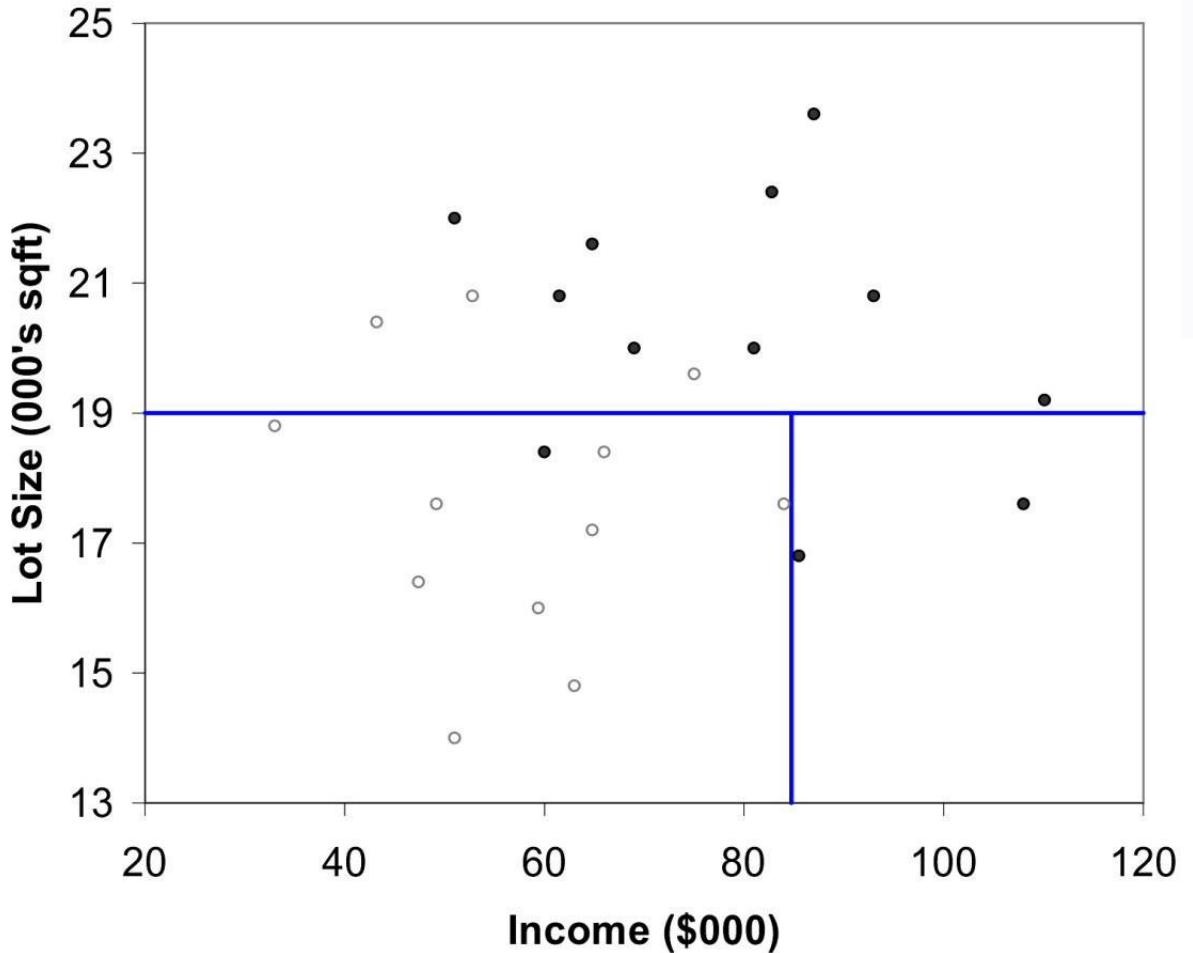
Household Number	Income (\$000s)	Lot Size (000s ft ²)	Ownership of Riding Mower
1	60.0	18.4	Owner
2	85.5	16.8	Owner
3	64.8	21.6	Owner
4	61.5	20.8	Owner
5	87.0	23.6	Owner
6	110.1	19.2	Owner
7	108.0	17.6	Owner
8	82.8	22.4	Owner
9	69.0	20.0	Owner
10	93.0	20.8	Owner
11	51.0	22.0	Owner
12	81.0	20.0	Owner
13	75.0	19.6	Nonowner
14	52.8	20.8	Nonowner
15	64.8	17.2	Nonowner
16	43.2	20.4	Nonowner
17	84.0	17.6	Nonowner
18	49.2	17.6	Nonowner
19	59.4	16.0	Nonowner
20	66.0	18.4	Nonowner
21	47.4	16.4	Nonowner
22	33.0	18.8	Nonowner
23	51.0	14.0	Nonowner
24	63.0	14.8	Nonowner



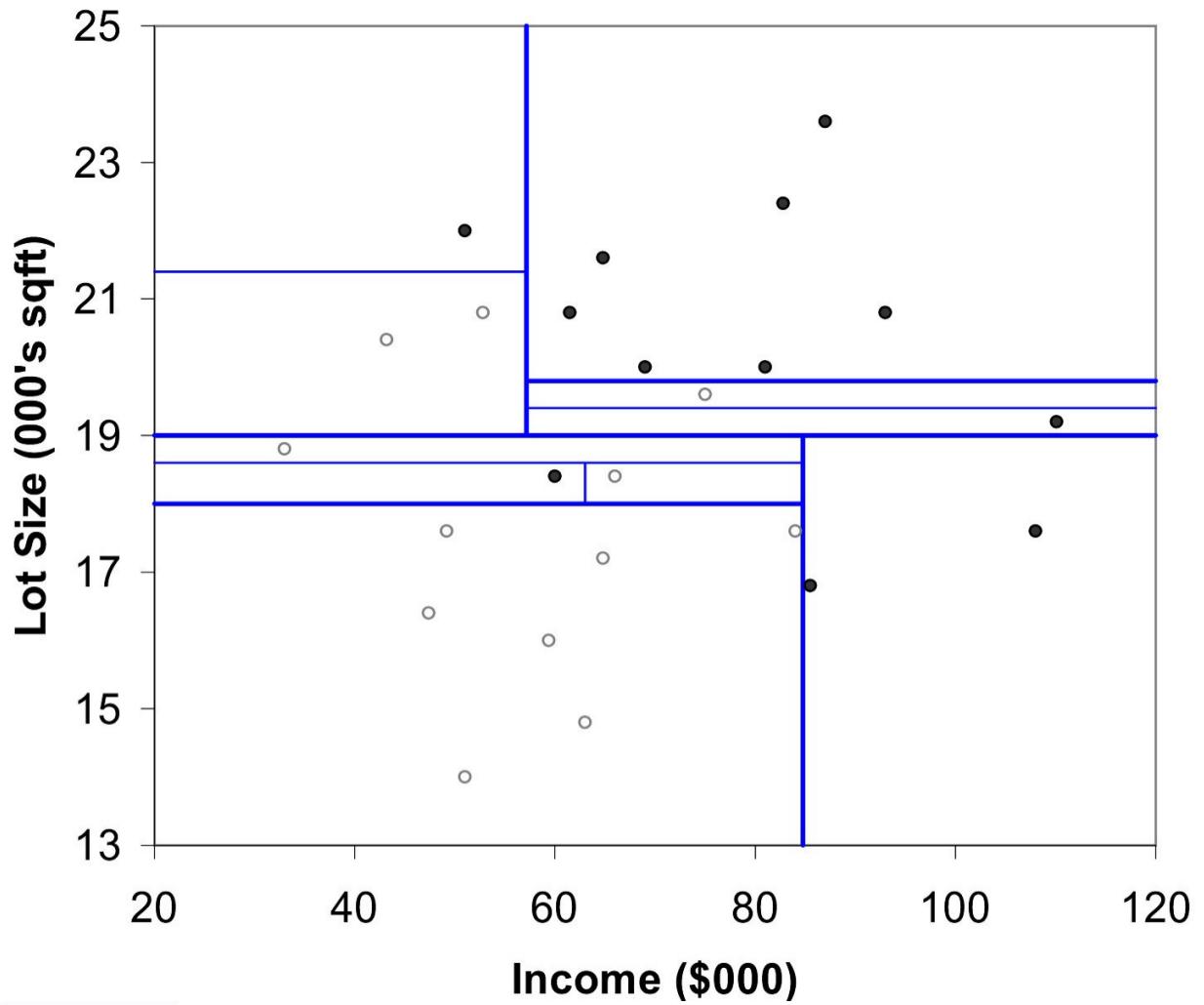
The first split: Lot Size = 19,000



Second Split: Income = \$84,000



After All Splits



Measuring Impurity or what split should I make?

Measuring impurity

- At each point we evaluate all possible splits.
- Decide on the split that gives the best gain at the moment.
- That is why it is called a *greedy* approach.

Measuring Impurity: regression

Measuring impurity

- Suppose we split a box into two smaller boxes.
- The decision is driven on the Residual Errors.
- Basically order all possible splits based on the average or sum of the errors and select the split that gives the lowest value.

Measuring Impurity: classification

How to determine the Best Split

- Greedy approach:
 - Nodes with **homogeneous** class distribution are preferred
 - Need a measure of node impurity:

C0: 5
C1: 5

Non-homogeneous,
High degree of impurity

C0: 9
C1: 1

Homogeneous,
Low degree of impurity

Gini Index

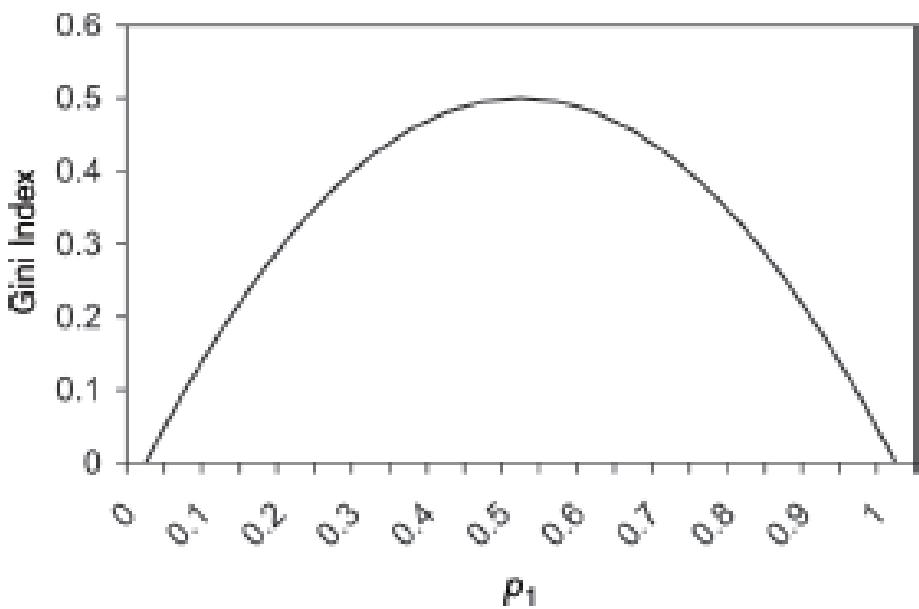
Gini Index for rectangle A containing m records

$$I(A) = 1 - \sum_{k=1}^m p_k^2$$

p = proportion of cases in rectangle A that belong to class k

- $I(A) = 0$ when all cases belong to same class
- Max value when all classes are equally represented (= 0.50 in binary case)

Gini Index



Examples for computing GINI

C1	0
C2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Gini} = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

C1	2
C2	4

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Gini} = 1 - (2/6)^2 - (4/6)^2 = 0.444$$

Splitting Based on GINI

- Used in CART, SLIQ, SPRINT.
- When a node p is split into k partitions (children), the quality of split is computed as,

$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child i,
 n = number of records at node p.

Entropy

$$\text{entropy}(A) = - \sum_{k=1}^m p_k \log_2(p_k)$$

p = proportion of cases (out of m) in rectangle A
that belong to class k

- Entropy ranges between 0 (most pure) and $\log_2(m)$ (equal representation of classes)

Examples for computing Entropy

C1	0
C2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Entropy} = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

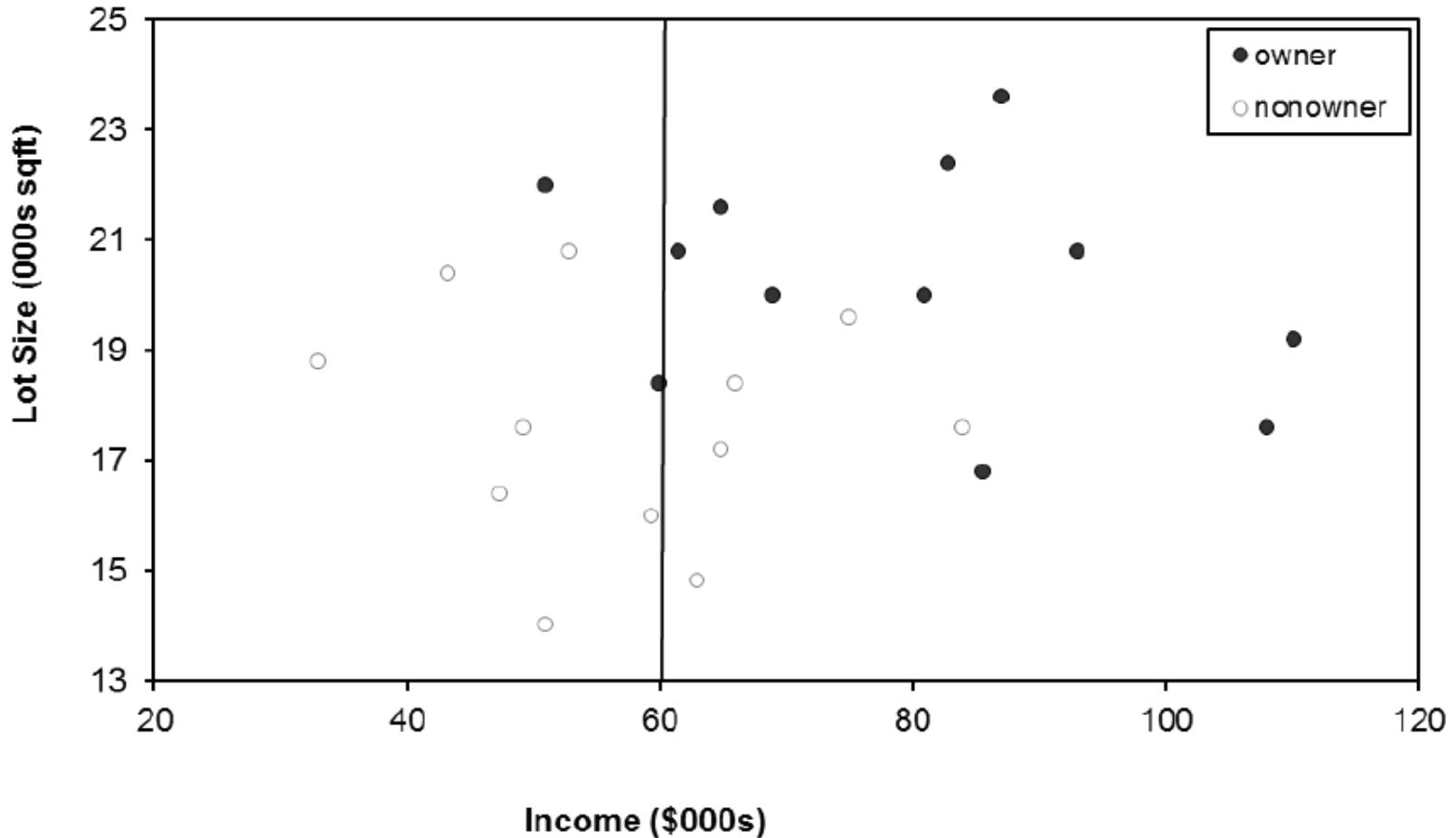
$$\text{Entropy} = -(1/6) \log_2 (1/6) - (5/6) \log_2 (1/6) = 0.65$$

C1	2
C2	4

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Entropy} = -(2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$$

Example



Example

$$\text{Gini_left} = 1 - (7/8)^2 - (1/8)^2 = 0.219$$

$$\text{entropy_left} = -(7/8) \log_2(7/8) - (1/8) \log_2(1/8) = 0.544$$

The right rectangle contains 11 owners and five nonowners. The impurity measures of the right rectangle are therefore

$$\text{Gini_right} = 1 - (11/16)^2 - (5/16)^2 = 0.430$$

$$\text{entropy_right} = -(11/16) \log_2(11/16) - (5/16) \log_2(5/16) = 0.896$$

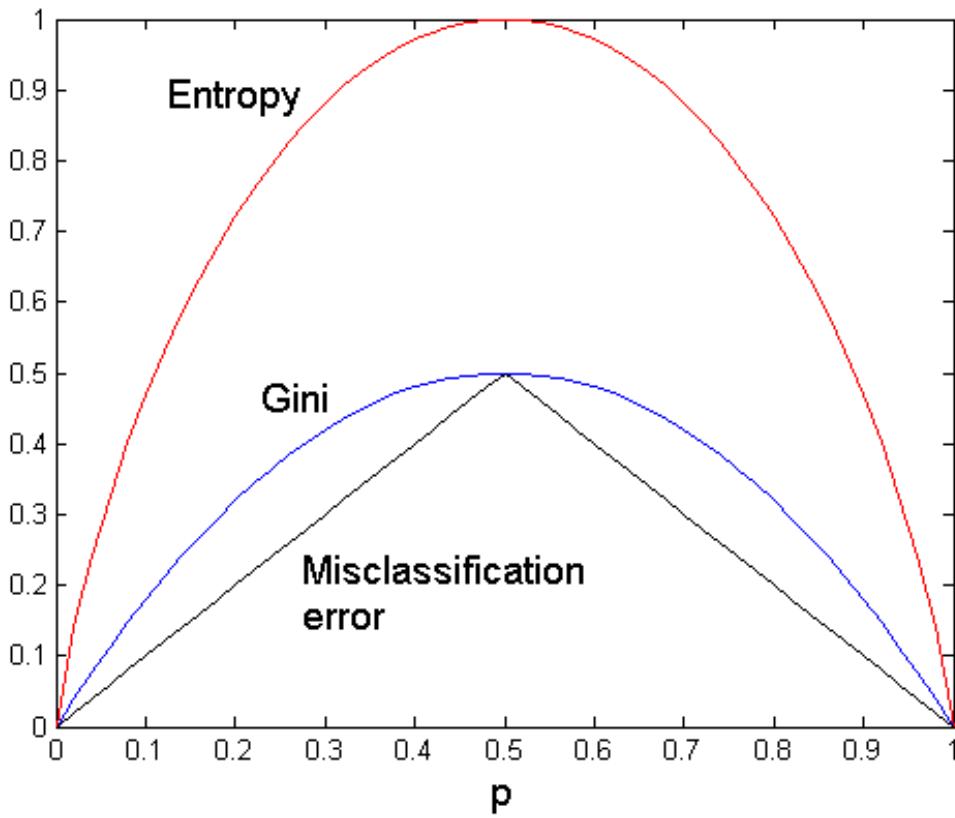
The combined impurity of the two rectangles that were created by the split is a weighted average of the two impurity measures, weighted by the number of records in each:

$$\text{Gini} = (8/24)(0.219) + (16/24)(0.430) = 0.359$$

$$\text{entropy} = (8/24)(0.544) + (16/24)(0.896) = 0.779$$

Comparison among Splitting Criteria

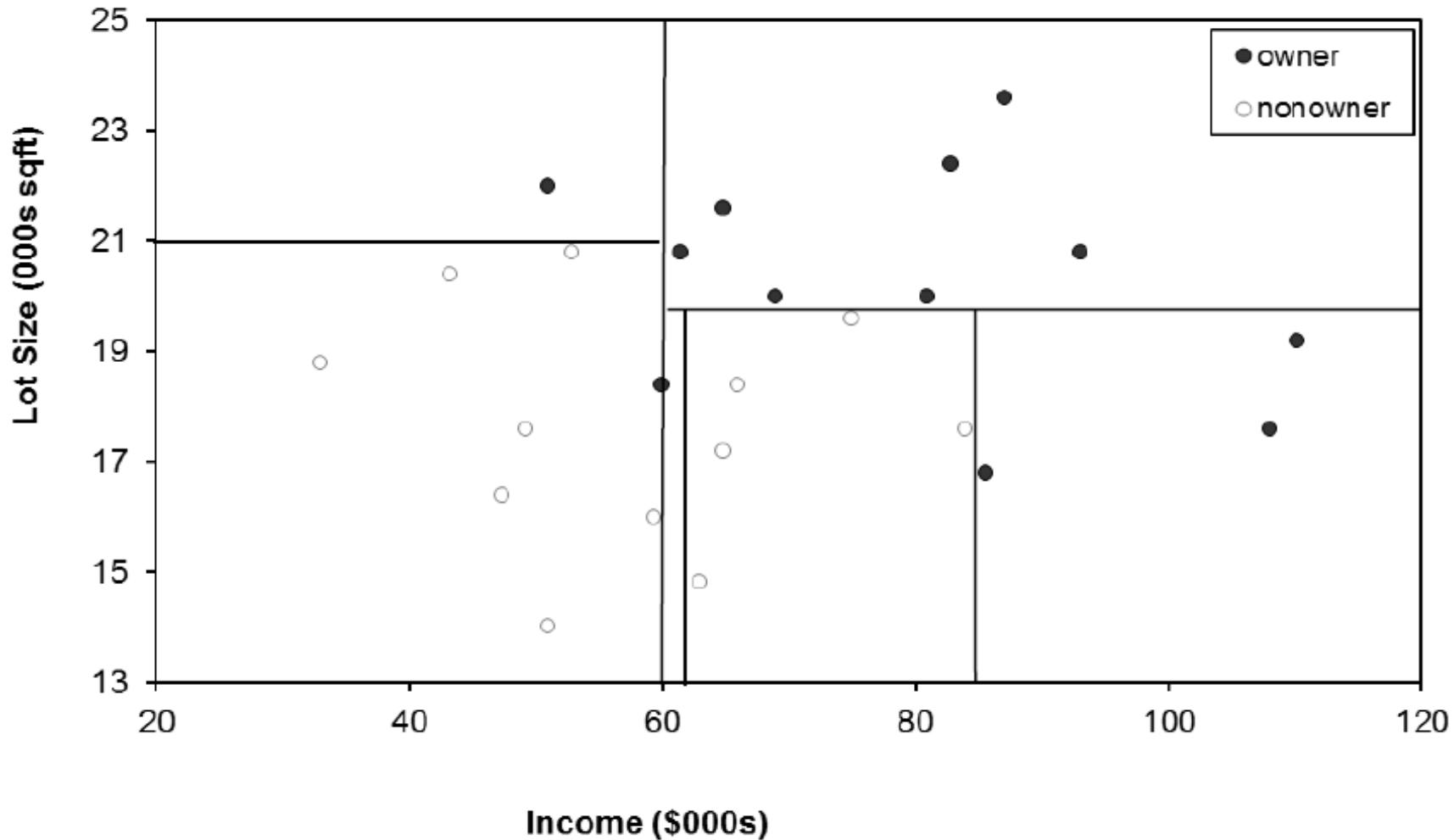
For a 2-class problem:



Impurity and Recursive Partitioning

- Obtain overall impurity measure (weighted avg. of individual rectangles)
- At each successive stage, compare this measure across all possible splits in all variables
- Choose the split that reduces impurity the most
- Chosen split points become nodes on the tree

Full grown tree – all boxes pure

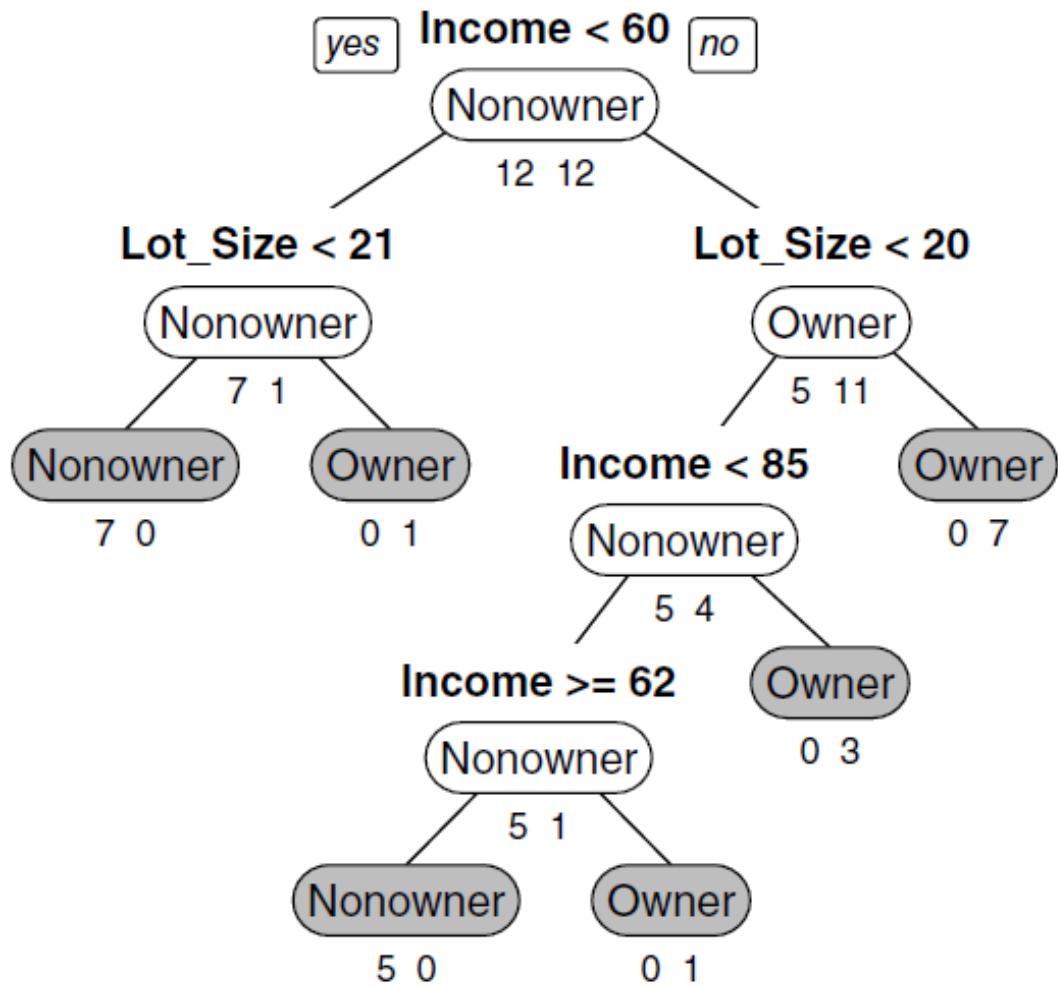


Example – R code

Library for trees

```
library(rpart)          → Library for trees  
library(rpart.plot)  
mower.df <- read.csv("RidingMowers.csv")  
  
# use rpart() to run a classification tree.  
# define rpart.control() in rpart() to determine the depth of the tree.  
class.tree <- rpart(Ownership ~ ., data = mower.df,  
                    control = rpart.control(maxdepth = 2), method = "class")  
## plot tree  
# use prp() to plot the tree. You can control plotting parameters such as color, shape,  
# and information displayed (which and where).  
prp(class.tree, type = 1, extra = 1, split.font = 1, varlen = -10)
```

Example



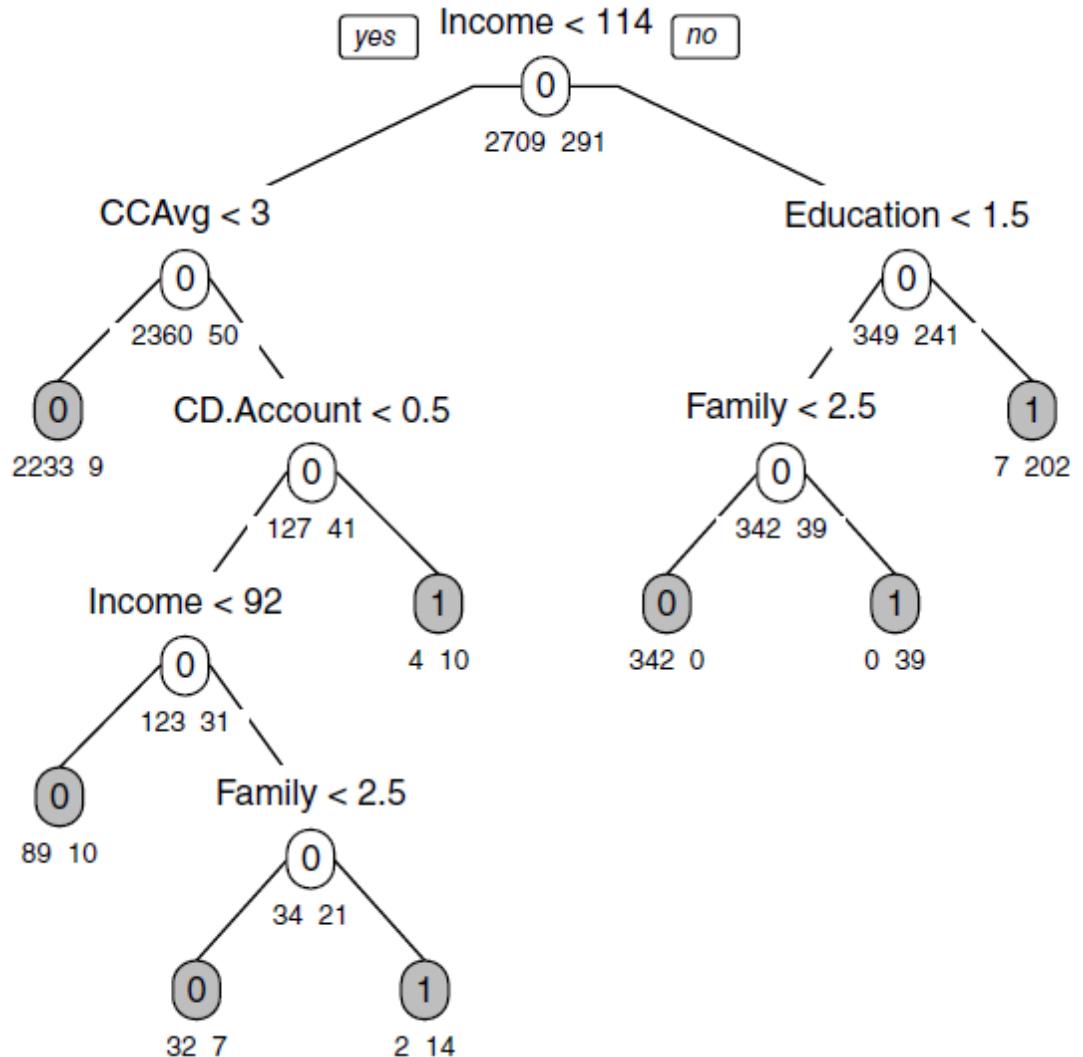
Avoid overfitting

TABLE 9.2

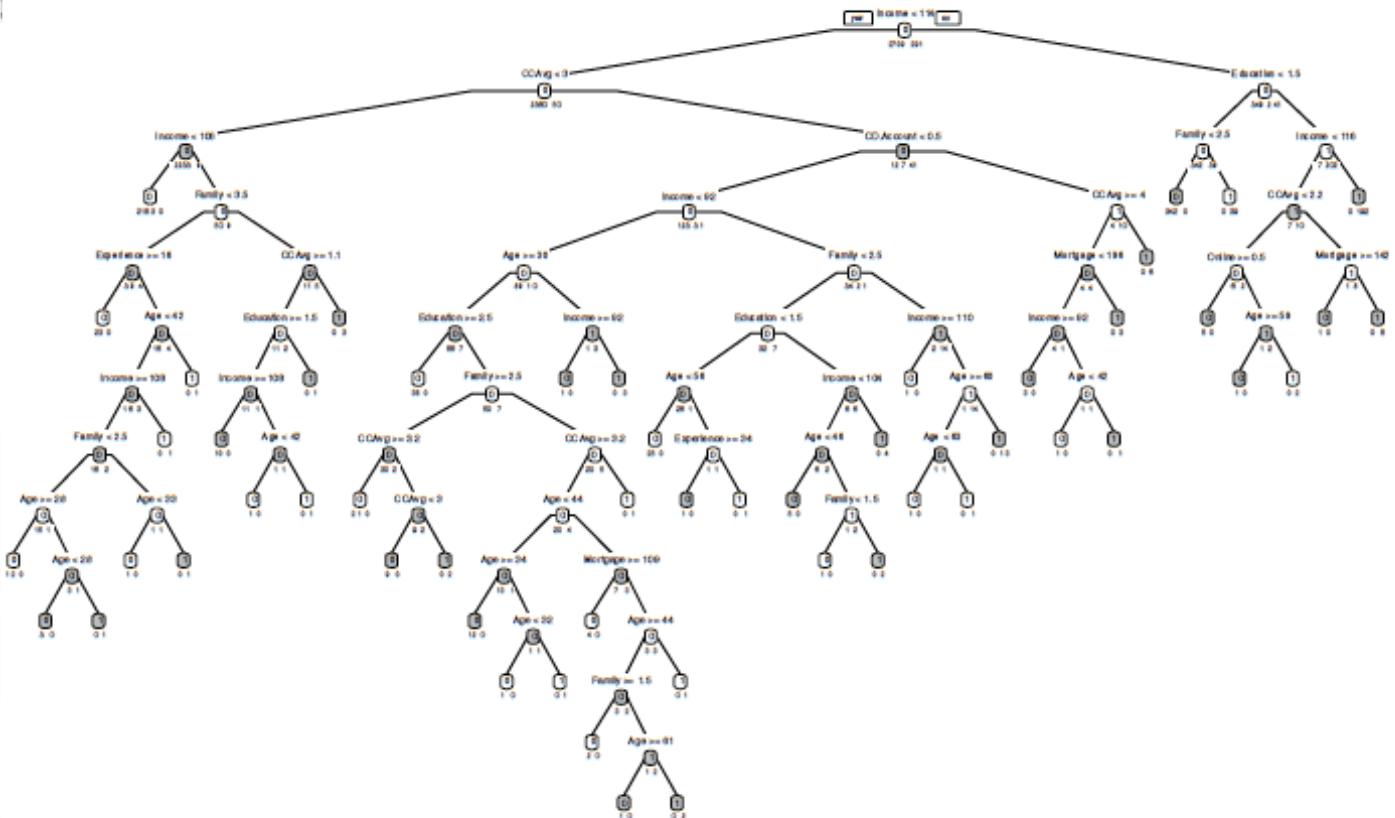
SAMPLE OF DATA FOR 20 CUSTOMERS OF UNIVERSAL BANK

ID	Age	Professional Experience	Income	Family Size	CC Avg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online Banking	Credit Card
1	25	1	49	4	1.60	UG	0	No	Yes	No	No	No
2	45	19	34	3	1.50	UG	0	No	Yes	No	No	No
3	39	15	11	1	1.00	UG	0	No	No	No	No	No
4	35	9	100	1	2.70	Grad	0	No	No	No	No	No
5	35	8	45	4	1.00	Grad	0	No	No	No	No	Yes
6	37	13	29	4	0.40	Grad	155	No	No	No	Yes	No
7	53	27	72	2	1.50	Grad	0	No	No	No	Yes	No
8	50	24	22	1	0.30	Prof	0	No	No	No	No	Yes
9	35	10	81	3	0.60	Grad	104	No	No	No	Yes	No
10	34	9	180	1	8.90	Prof	0	Yes	No	No	No	No
11	65	39	105	4	2.40	Prof	0	No	No	No	No	No
12	29	5	45	3	0.10	Grad	0	No	No	No	Yes	No
13	48	23	114	2	3.80	Prof	0	No	Yes	No	No	No
14	59	32	40	4	2.50	Grad	0	No	No	No	Yes	No
15	67	41	112	1	2.00	UG	0	No	Yes	No	No	No
16	60	30	22	1	1.50	Prof	0	No	No	No	Yes	Yes
17	38	14	130	4	4.70	Prof	134	Yes	No	No	No	No
18	42	18	81	4	2.40	UG	0	No	No	No	No	No
19	46	21	193	2	8.10	Prof	0	Yes	No	No	No	No
20	55	28	21	1	0.50	Grad	0	No	Yes	No	No	Yes

Avoid overfitting – two tree examples



Avoid overfitting – two tree examples



Avoid overfitting – two tree examples

```
> # default tree: training  
> confusionMatrix(default.ct.poin  
Confusion Matrix and Statistics
```

		Reference
Prediction	0	1
0	2696	26
1	13	265

Accuracy : 0.987

```
> # default tree: validation  
> confusionMatrix(default.ct.poin  
Confusion Matrix and Statistics
```

		Reference
Prediction	0	1
0	1792	18
1	19	171

Accuracy : 0.9815

```
> # deeper tree: training  
> confusionMatrix(deeper.ct.point.pre  
Confusion Matrix and Statistics
```

		Reference
Prediction	0	1
0	2709	0
1	0	291

Accuracy : 1

```
> # deeper tree: validation  
> confusionMatrix(deeper.ct.point.pre  
Confusion Matrix and Statistics
```

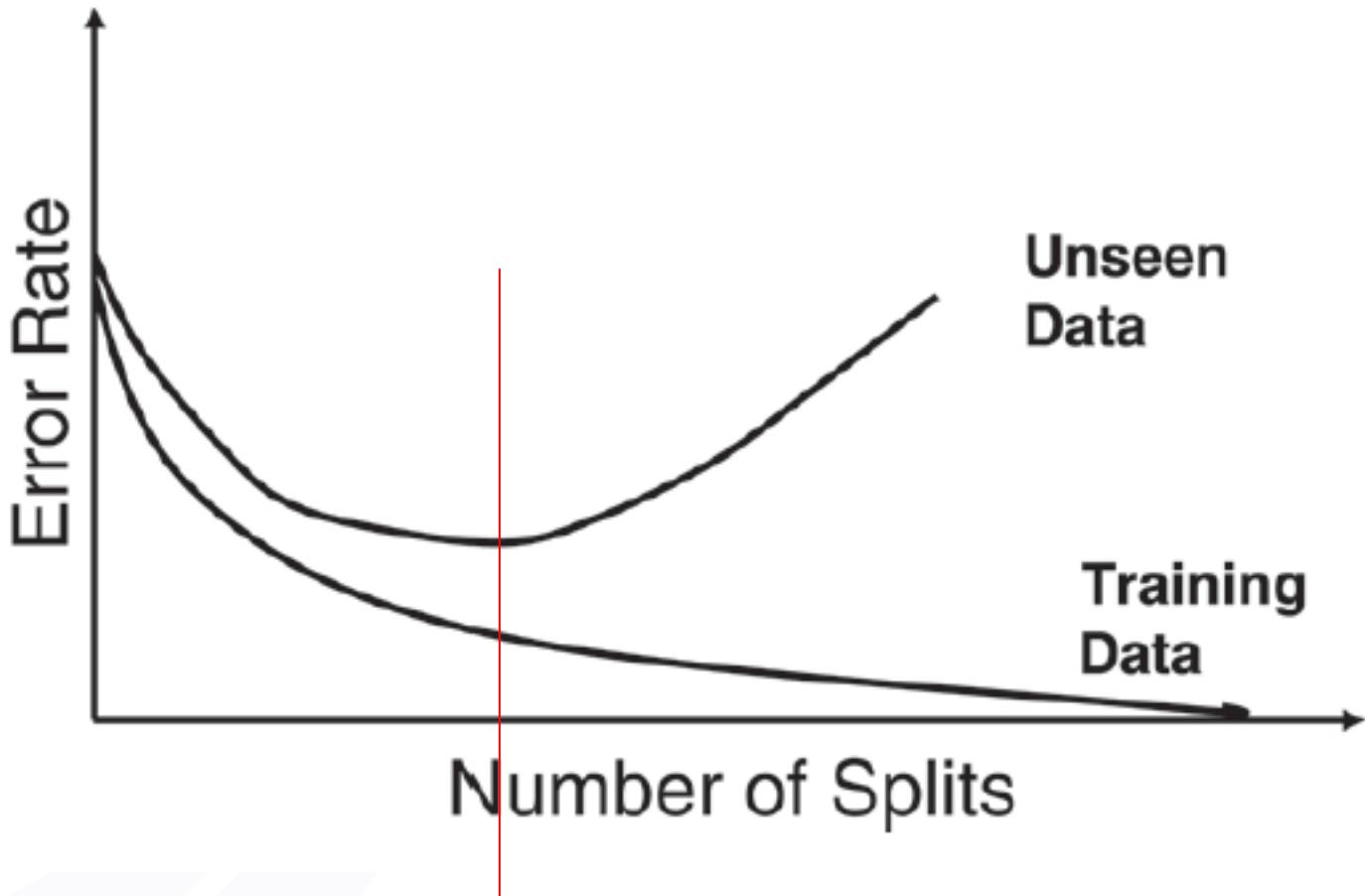
		Reference
Prediction	0	1
0	1788	19
1	23	170

Accuracy : 0.979

Stopping Tree Growth

- Natural end of process is 100% purity in each leaf
- This **overfits** the data, which end up fitting noise in the data
- Overfitting leads to low predictive accuracy of new data
- Past a certain point, the error rate for the validation data starts to increase

Ideal tree



Pruning

- CART lets tree grow to full extent, then prunes it back.
- Idea is to find that point at which the validation error begins to rise, using the validation data.
- Generate successively smaller trees by pruning leaves.
- At each pruning stage, multiple trees are possible.
- Use *cost complexity* to choose the best tree at that stage.

Cost Complexity

$$CC(T) = Err(T) + a * L(T)$$

$CC(T)$ = cost complexity of a tree

$Err(T)$ = proportion of misclassified records

$L(T)$ = # of terminal nodes on the tree

a = penalty factor attached to tree size (set by user)

- Among trees of given size, choose the one with lowest CC
- Do this for each size of tree

Using Validation Error to Prune

Pruning process yields a set of trees of different sizes and associated error rates

Two trees of interest:

- Minimum error tree
 - Has lowest error rate on validation data
- Best pruned tree
 - Smallest tree within one std. error of min. error This adds a bonus for simplicity/parsimony

Guide to build a tree

1. Partition the data into training and validation sets.
2. Grow the tree with the training data.
3. Prune it successively, step by step, recording CP (using the *training* data) at each step.
4. Note the CP that corresponds to the minimum error on the *validation* data.
5. Repartition the data into training and validation, and repeat the growing, pruning and CP recording process.
6. Do this again and again, and average the CP's that reflect minimum error for each tree.
7. Go back to the original data, or future data, and grow a tree, stopping at this optimum CP value.

Guide to build a tree - R

```
# argument xval refers to the number of folds to use in rpart's built-in  
# cross-validation procedure  
# argument cp sets the smallest value for the complexity parameter.  
cv.ct <- rpart(Personal.Loan ~ ., data = train.df, method = "class",  
    cp = 0.00001, minsplit = 5, xval = 5)  
# use printcp() to print the table.  
printcp(cv.ct)
```

Output

	CP	nsplit	rel error	xerror	xstd
1	0.3350515	0	1.000000	1.000000	0.055705
2	0.1340206	2	0.329897	0.37457	0.035220
3	0.0154639	3	0.195876	0.19931	0.025917
4	0.0068729	7	0.134021	0.17182	0.024096
5	0.0051546	12	0.099656	0.17182	0.024096
6	0.0034364	14	0.089347	0.16838	0.023858
7	0.0022910	19	0.072165	0.17182	0.024096
8	0.0000100	25	0.058419	0.17182	0.024096

Error

Standard error

Best parsimonious tree

Best tree

Regression Trees

Regression Trees for Prediction

- Used with continuous outcome variable
- Procedure similar to classification tree
- Many splits attempted, choose the one that minimizes impurity

Differences from CT

- Prediction is computed as the **average** of numerical target variable in the rectangle (in CT it is majority vote).
- Impurity measured by **sum of squared deviations** from the mean.
- Performance measured by RMSE (root mean squared error).

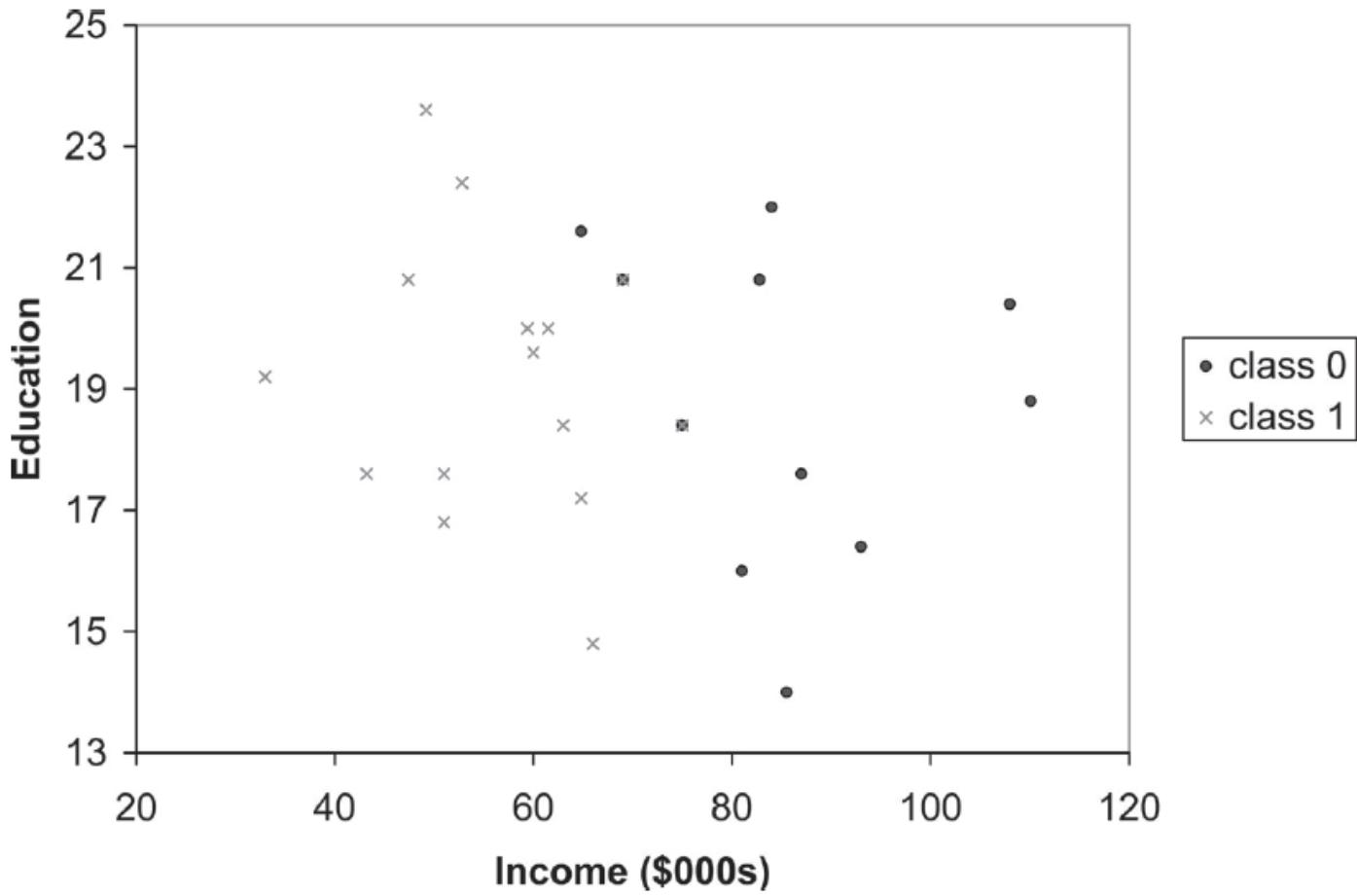
Advantages of trees

- Easy to use, understand
- **Produce rules that are easy to interpret & implement**
- Variable selection & reduction is automatic
- Do not require the assumptions of statistical models
- Can work without extensive handling of missing data
- Efficient to use

Disadvantages

- May not perform well where there is structure in the data that is not well captured by horizontal or vertical splits
- Since the process deals with one variable at a time, no way to capture interactions between variables
- Costly to build
- Quite unstable, require cross validation

Disadvantages



Summary

- Classification and Regression Trees are an easily understandable and transparent method for predicting or classifying new records.
- A tree is a graphical representation of a set of rules.
- Trees must be pruned to avoid over-fitting of the training data.
- As trees do not make any assumptions about the data structure, they usually require large samples.

Summary

"Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani
"

Combining Methods

Why Combine?

- Ensemble of methods often predicts more accurately.
- Business goal may require multiple methods.

Netflix Prize



Leaderboard

Showing Test Score. [Click here to show quiz score](#)

Display top 20 leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43



Ensembles

- Combines prediction from
 - 1. Different methods
 - 2. Different approaches
 - 3. Different data

Popular Ensemble methods

- Bagging : averaging the prediction over a collection of classification
- Boosting : weighted vote the with a collection of classifiers
- Ensemble : combining a set of heterogeneous of classifiers

Bagging (= bootstrap aggregating)

The “multiplier” effect in bagging comes from multiple bootstrap samples, rather than multiple methods. Bootstrapping is to take resamples, with replacement, from the original data.

1. Generate multiple bootstrap resamples
2. Run algorithm on each and produce scores
3. Average those scores (or take majority vote)



Random Forests

- Often a single decision tree provides a model too simple or too specific.
- Many models working together are better than one model doing it all.
- A random forest builds hundreds of decision trees and combines them into a single model (also known as ensemble).

How it works

- Build each decision tree to its maximum depth (no pruning) and over fit the data for each tree.
- The overfitting is outweighed by the multiple trees that use different variables and fit the data differently.
- Random forest uses randomness to select observations and variables (two levels of randomness).

How it works

- Aggregate decisions into one final model.
- Simple majority rules or a weighed score corresponding to the quality of each tree.
- Often build 100 to 500 trees.

Example

- Will it rain tomorrow?
- We built 100 decision trees.
- All trees are treated as equals.
- If 80 out of 100 predict that it will rain tomorrow, then the final prediction is that it will rain tomorrow.

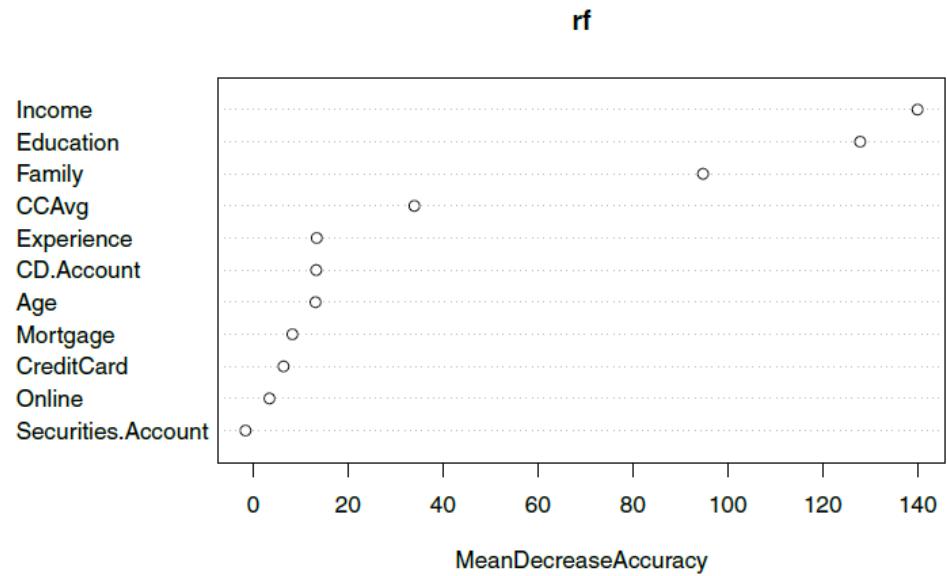
How it works

- At the end we lose explanatory power.
- Harder to explain results.
- Cannot convey to rules.
- However...

How it works

```
library(randomForest)
## random forest
rf <- randomForest(as.factor(Personal.Loan) ~ ., data = train.df, ntree = 500,
                     mtry = 4, nodesize = 5, importance = TRUE)

## variable importance plot
varImpPlot(rf, type = 1)
```



How it works

- ... and that is why we can use it for variable selection.

Advantages

- Very robust to noise.
- That means that any changes to the dataset will have minimal effect to the final decision.
- Generally very competitive with non-linear classifiers as NN and SVM.
- Random forests manage unbalanced data very well (ex: binary classification will one class having only 5% of the dataset).

Advantages

- Little pre-processing of the dataset (data don't need normalization).
- No need for variable selection.
- Since trees use two levels of randomness, each tree is an independent model and the resulting forest does not usually over-fit the dataset.

Boosting

Iteratively focus attention on the records that are misclassified, or where error is greatest

1. Fit model to data
2. Resample records with highest weights to misclassified or highest errors
3. Fit model to new sample
4. Repeat steps 2-3

Ensembles summary

Ensembles...

1. Generally perform better than individual models
2. Have many variants (averaging, weighted averaging, voting, medians, resampling)
3. Facilitate "**parallel processing**," e.g. in contests where multiple teams' models can be combined
4. Help **mitigate overfitting** (but do not cure it)
5. Are **black-box** – transparent methods like trees lose transparency when ensembled

Summary

"Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani
"

Support vector machines

Ilias Thomas

What are SVM?

- SVMs are supervised machine learning models which are capable of analyzing and recognizing patterns.
- They can achieve both classification and regression.
- SVM was first introduced by Vladimir Vapnik and Corinna Cortes in 1995 while working at AT&T.
- SVM is a machine learning method derived from Statistical Learning Theory (SLT)
- SVM becomes famous and popular because of its success in handwritten digit recognition (1.1% test error rate for SVM)

Introduction

What is a hyperplane?

- In a p -dimensional space, a hyperplane is a flat affine subspace of hyperplane dimension $p-1$.
- For instance, in two dimensions, a hyperplane is a flat one-dimensional subspace—in other words, a line.
- In three dimensions, a hyperplane is a flat two-dimensional subspace—that is, a plane.

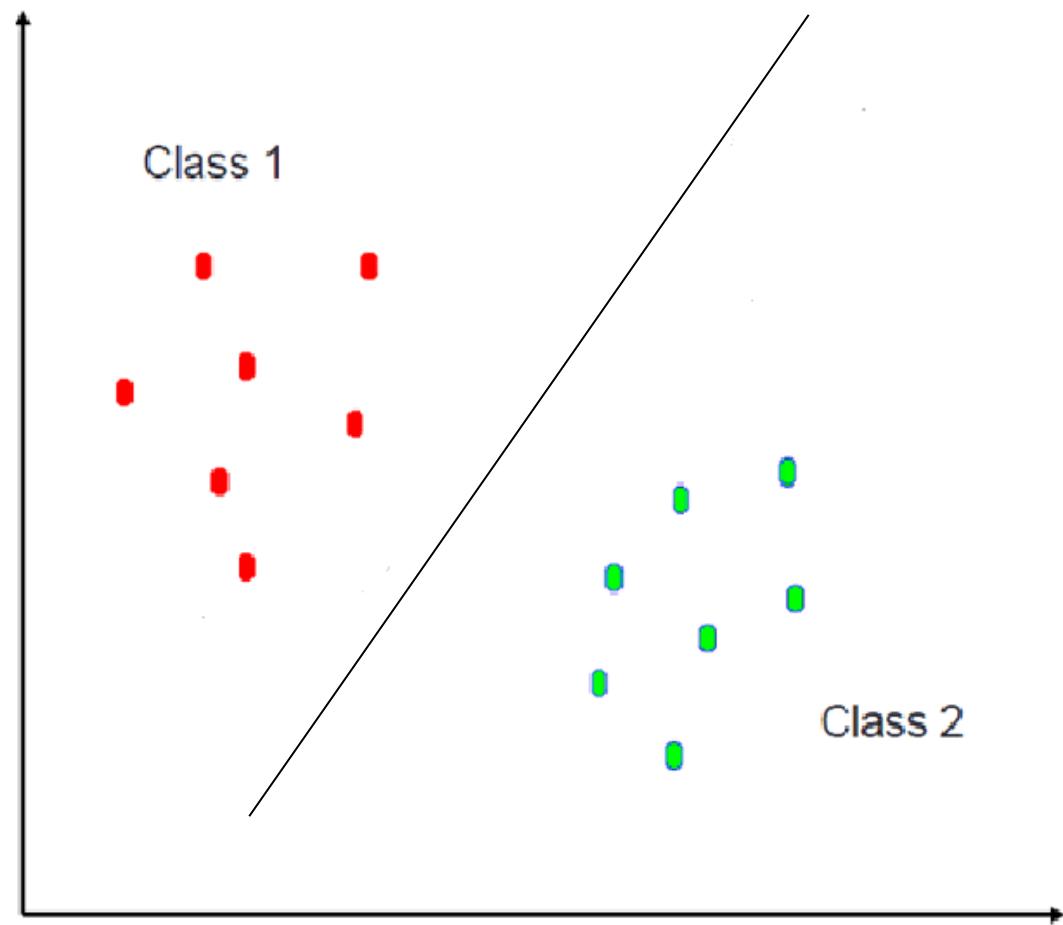
What is a hyperplane?

- For instance, in two dimensions, a hyperplane is a flat one-dimensional subspace—in other words, a line.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0 \quad (1)$$

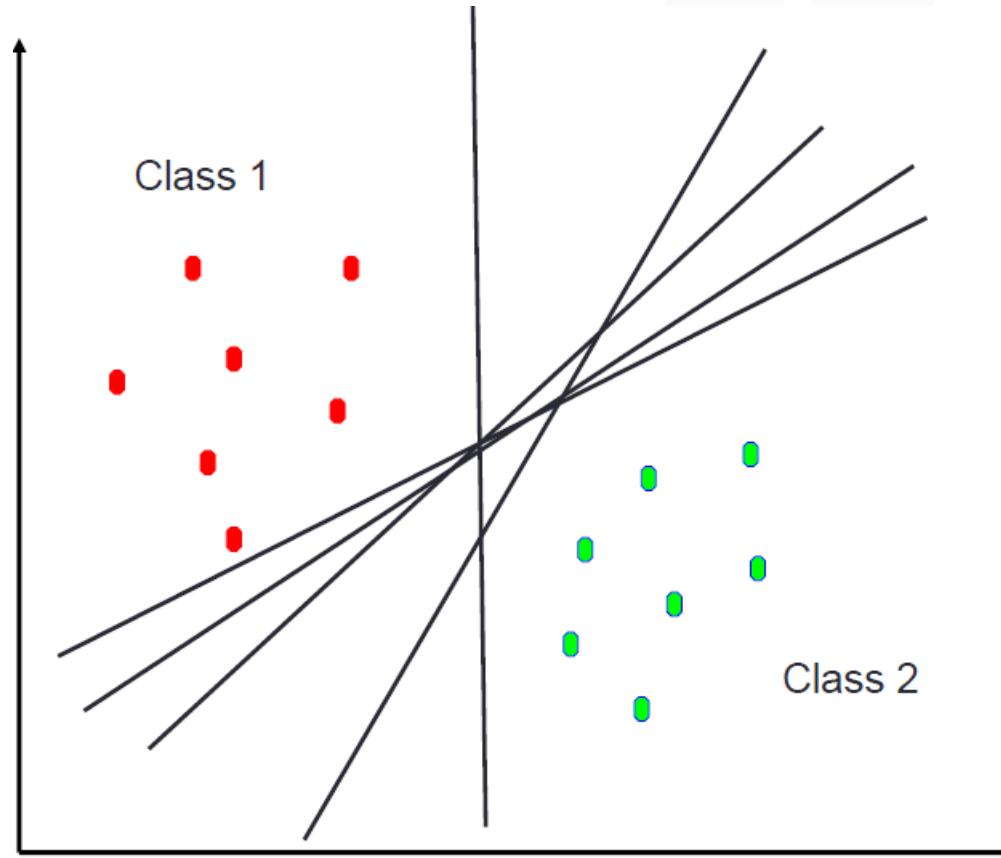
- If we think about it in the sense of a classifier then a point exactly on the line will solve the equation.
- A point in either side of the line will give results that are either positive or negative.

Example



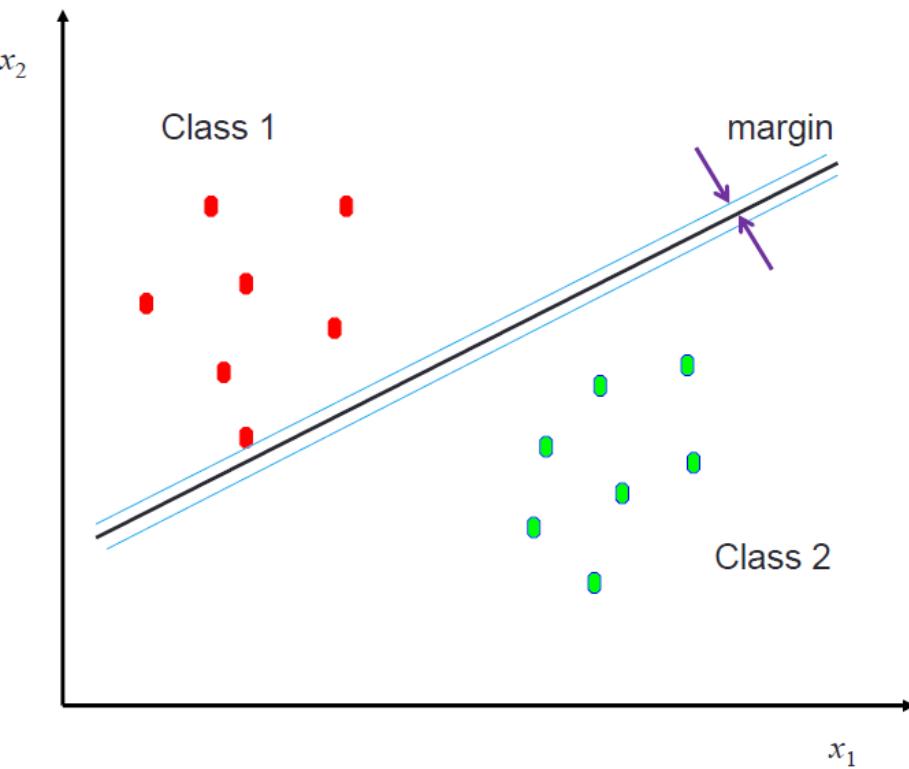
Explore the Problem

- Is there any advantage of choosing one line over the other?
- Which Line is the best for classification?



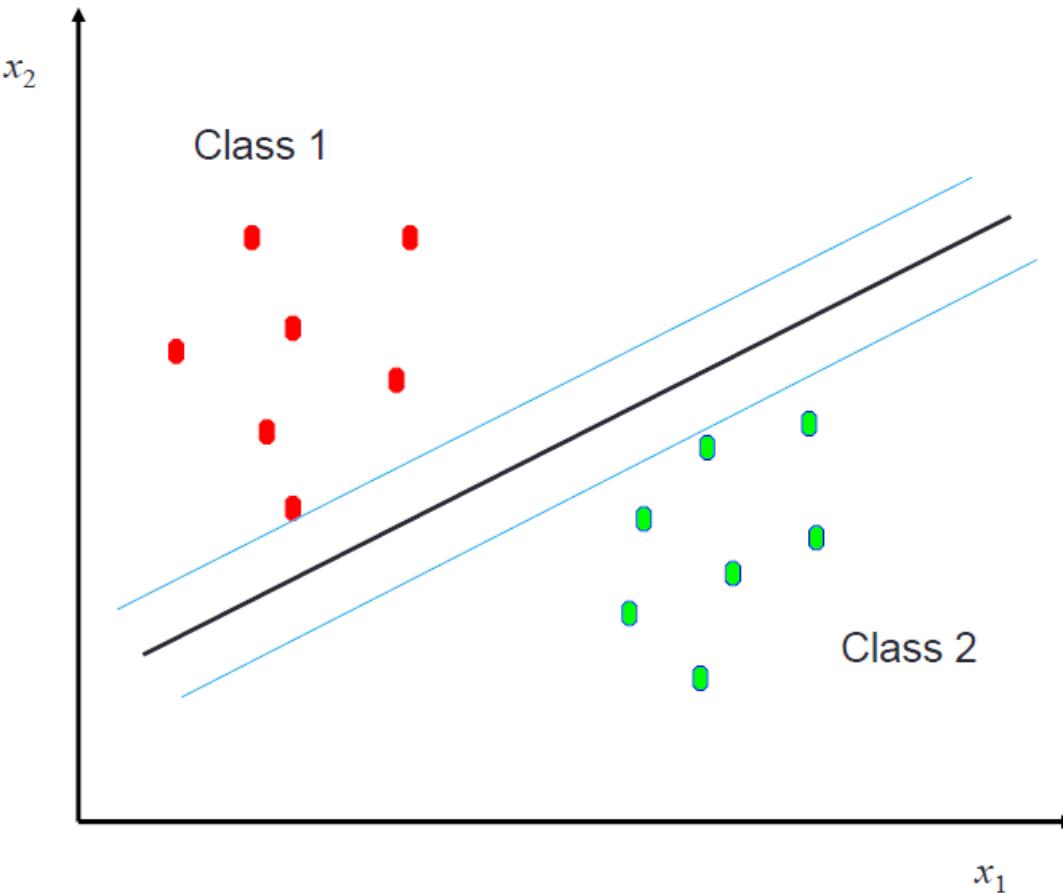
A line with narrow margin

- A line which is close to one of the classes leaves us with small margins (Blue lines).
- If data is noisy, classification error may happen easily.



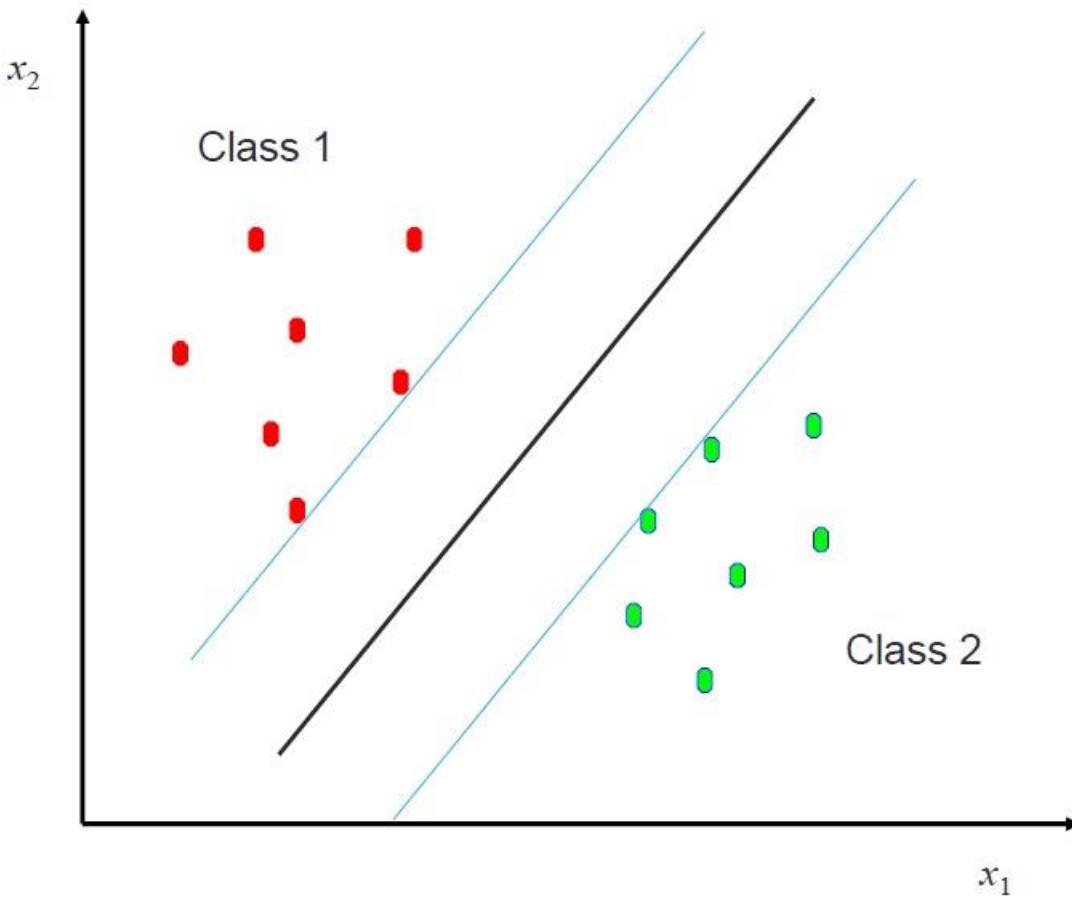
A line with wider margin works better

- When we choose a separation line which generates a wider margin, classification error can be reduced



Is this the best line?

- We got a line with a wider margin.
- Is this the optimum line for this problem?



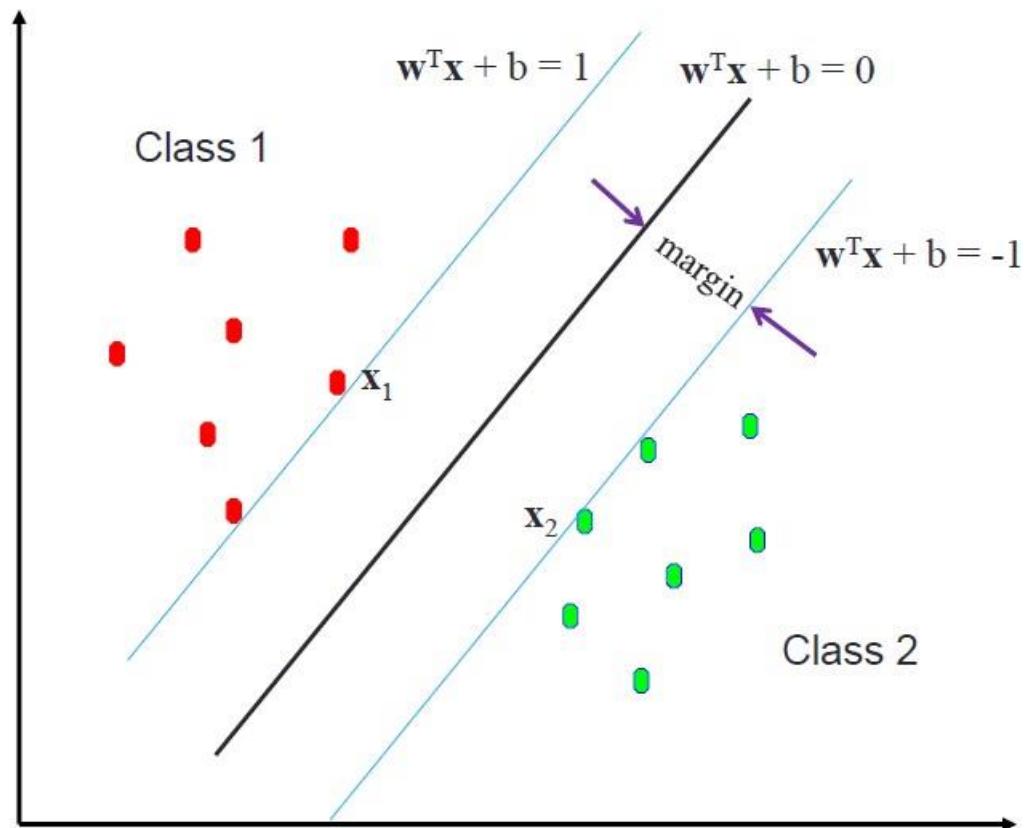
Maximal margin classifier

- Maximizing the margin seems good because points near the decision surface represent very uncertain classification decisions.
- This means that there is almost a 50% chance of the classifier deciding either way.
- A classifier with a large margin makes no low certainty classification decisions.
- This gives you a classification safety margin: a slight error in measurement or a slight document variation will not cause a misclassification.

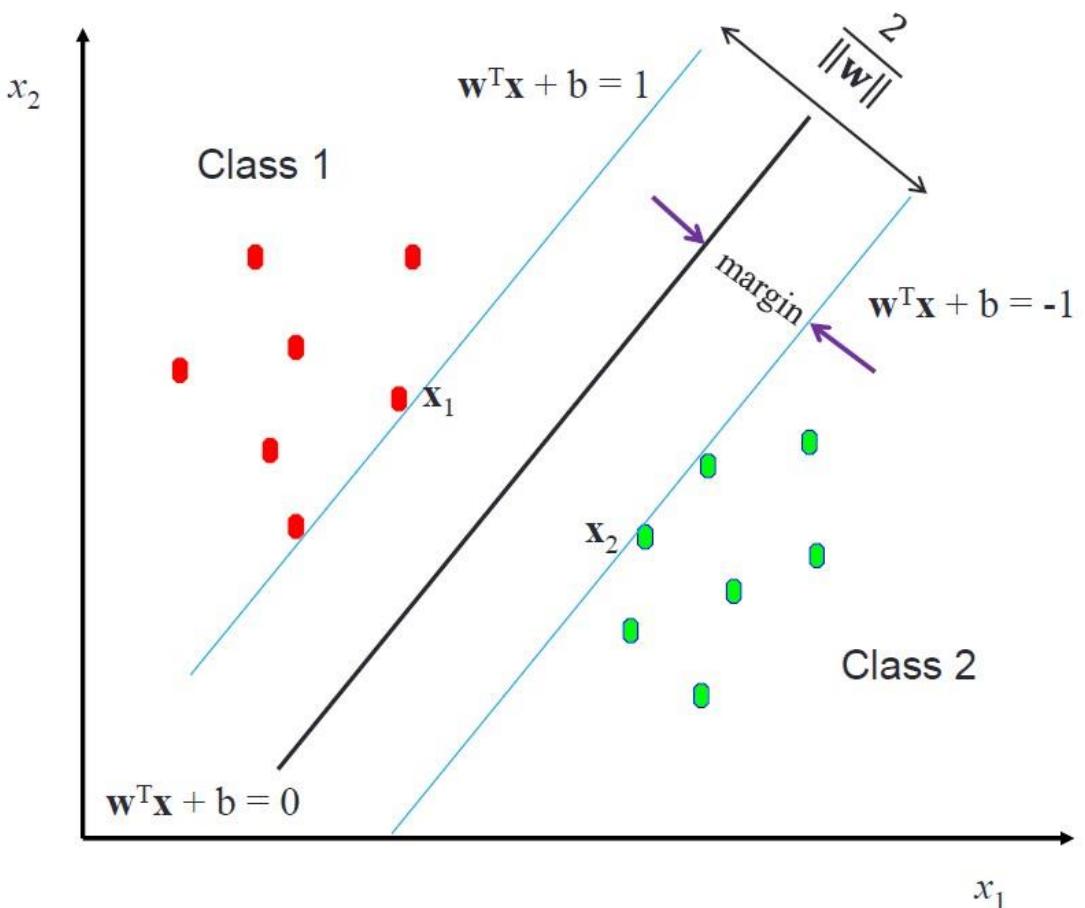
The hyperplane

Remarks:

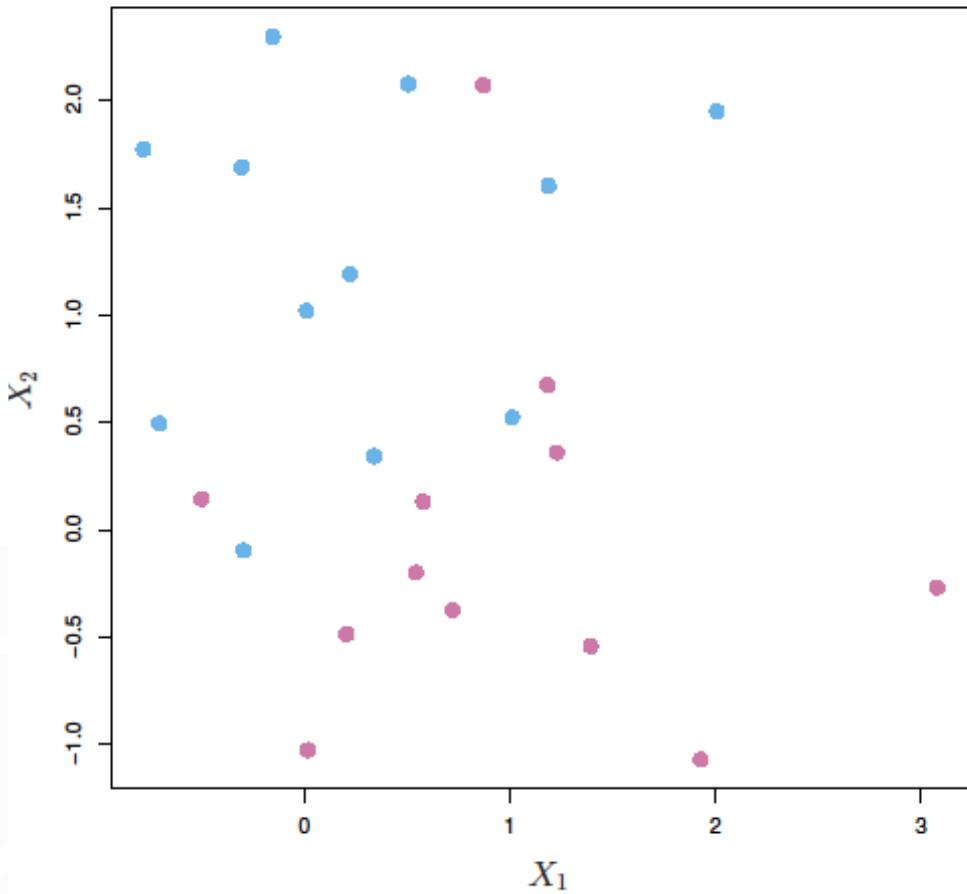
1. The patterns located on margin boundary (x_1 and x_2) are called support vectors.
2. The target value of class1 is +1 and for class2 is -1.



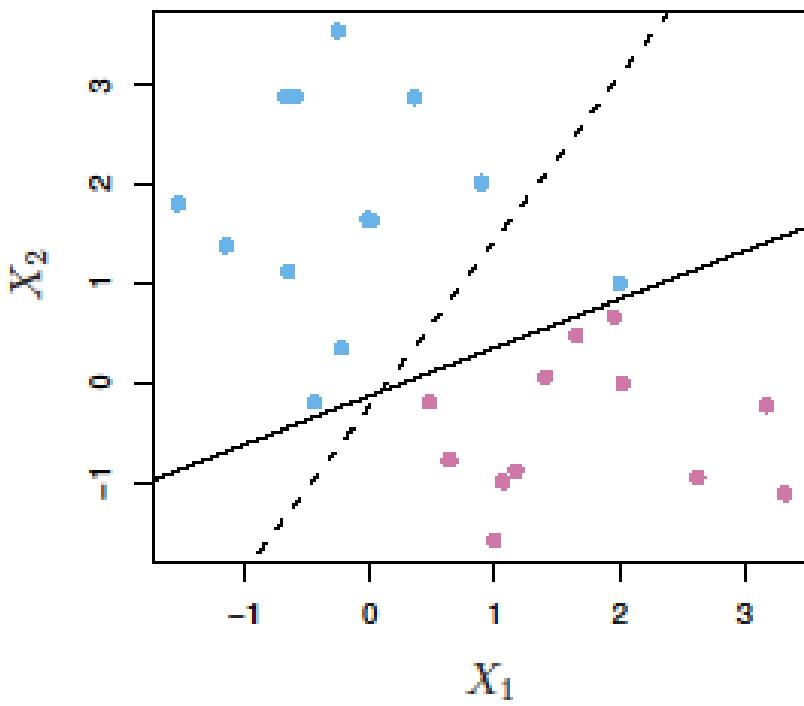
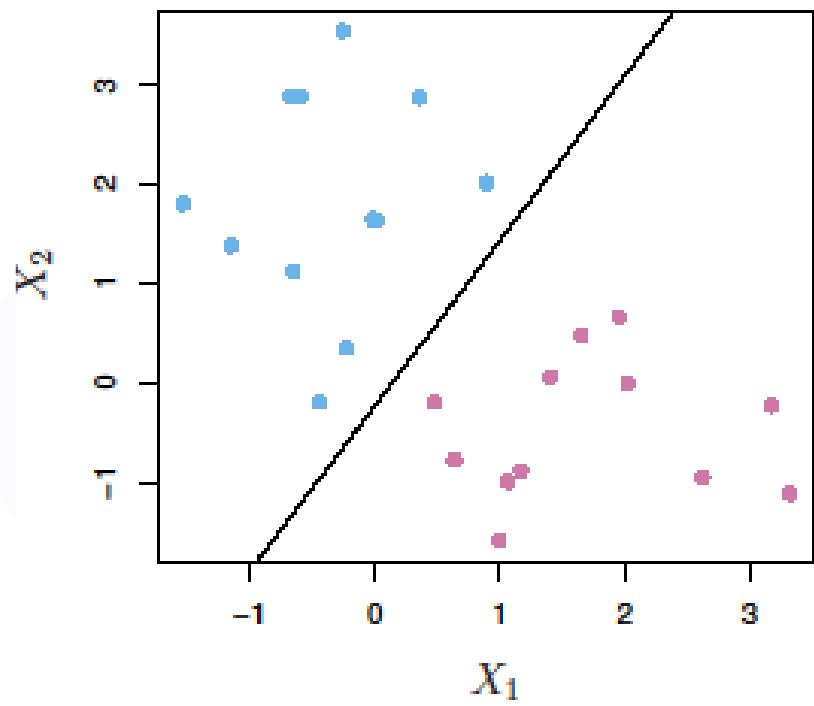
Maximum distance between two classes



Support vector classifier

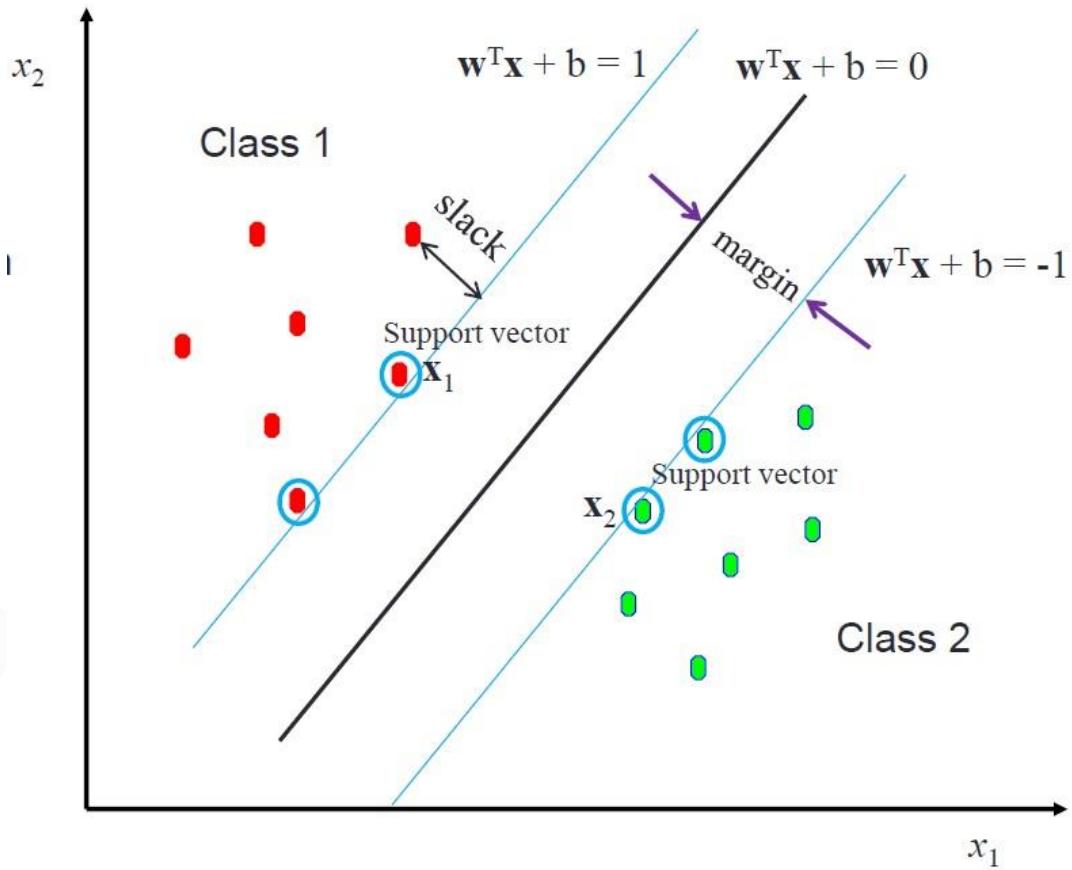


Support vector classifier

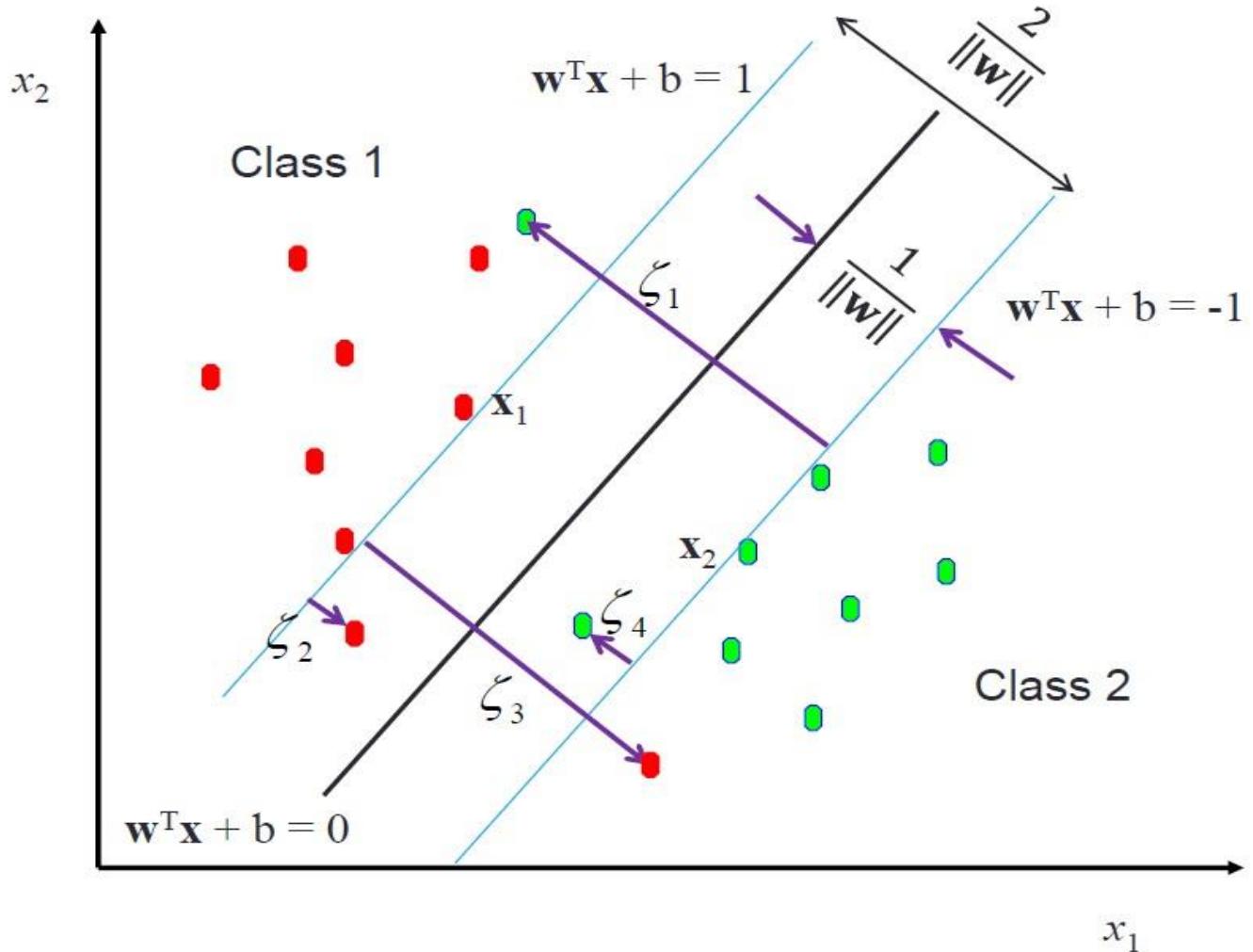


The slack

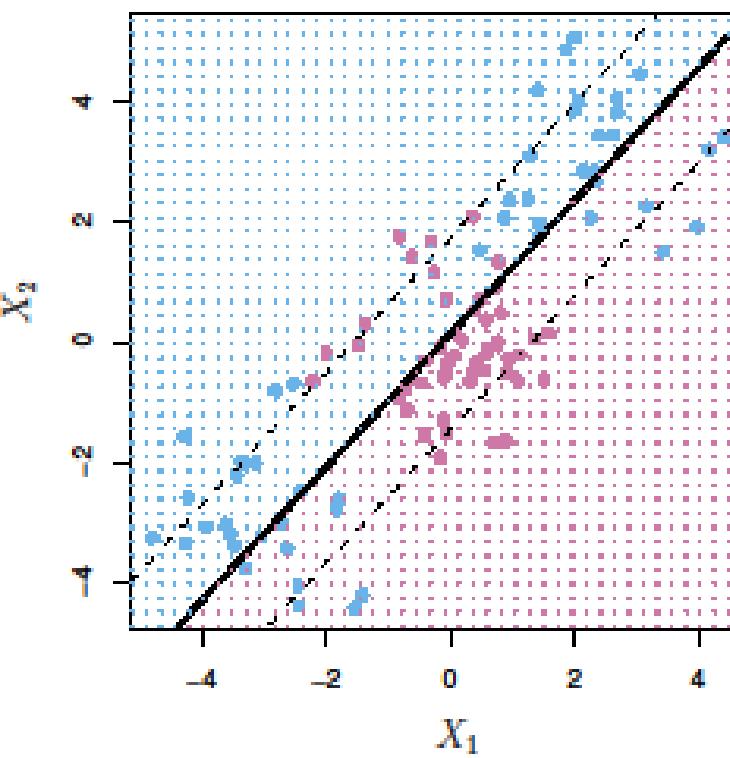
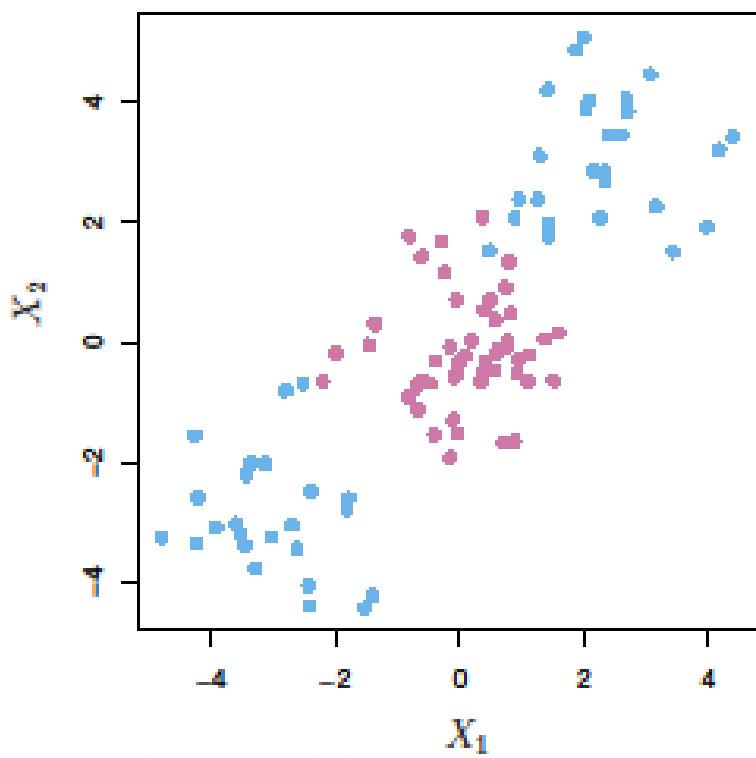
When the slack is zero, the vector is located on the margin boundary. In this case, it is called support vector, because it supports the plane of separation.



The slack again



How about this case?



What is a kernel?

- A kernel function is a function that embeds the data into a feature space where the nonlinear pattern appears linear.
- It returns the inner product in some feature space.
- If $\Phi: \mathbf{x} \rightarrow \phi(\mathbf{x})$, the inner product becomes: $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$
- The kernel function can be defined by

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

- K is called a kernel function.
- K can be thought of as a similarity measure between two patterns x and y.
- In Kernel functions the dot products can be replaced with the kernel.

$$\langle \mathbf{x}, \mathbf{y} \rangle \leftarrow K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

Types of kernels

1. Linear Kernel

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle$$

2. Polynomial Kernel

$$K(\mathbf{x}, \mathbf{z}) = (\gamma \langle \mathbf{x}, \mathbf{z} \rangle + r)^d$$

Where γ , r ; and d are the kernel parameters and $\gamma > 0$

3. Gaussian Radial Basis Function

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2)$$

- The GRBF kernel non-linearly maps data into a higher dimensional feature space
- It is better to select a kernel with the number of kernel parameters as small as possible, which GRBF fulfils.
- The RBF kernel has fewer numerical difficulties

Types of kernels

4. Exponential Radial Basis Function

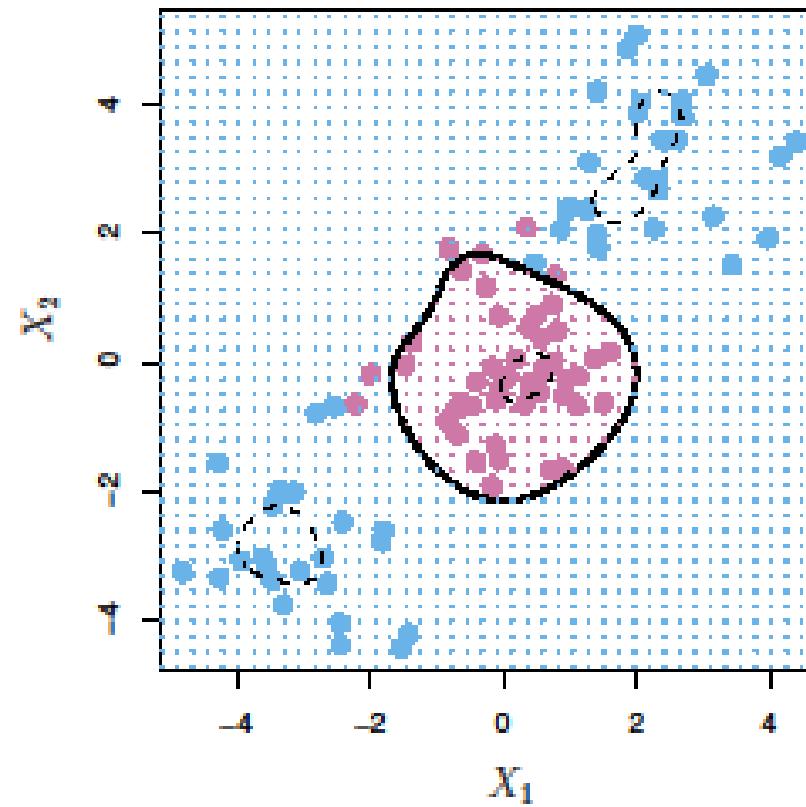
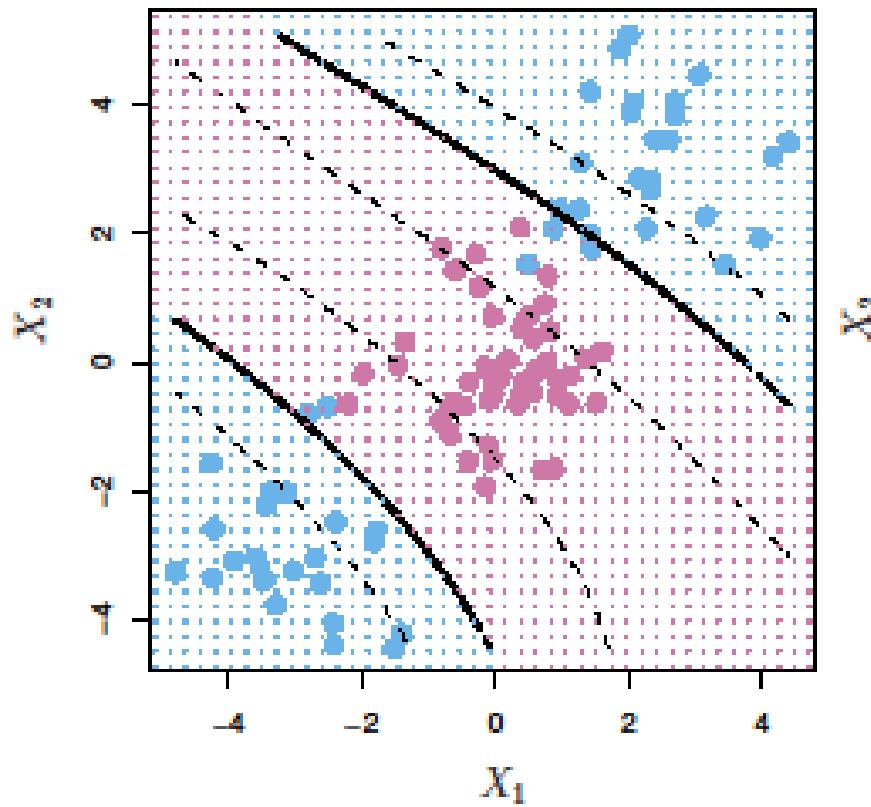
$$K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|)$$

5. Sigmoid

A SVM model using a sigmoid kernel function is equivalent to a two-layer, feed forward neural network,

$$K(\mathbf{x}, \mathbf{z}) = \tanh(\gamma \langle \mathbf{x}, \mathbf{z} \rangle + r)$$

Support vector machines



Multiple Classes

- One against all
- The “one against all” strategy consists of constructing one SVM per class, which is trained to distinguish the samples of one class from the samples of all remaining classes.
- Usually, classification of an unknown pattern is done according to the maximum output among all SVMs
- One against one
- The “one against one” strategy consists in constructing one SVM for each pair of classes.
- Thus, for a problem with c classes, $c(c-1)/2$ SVMs are trained to distinguish the samples of one class from the samples of another class.
- Usually, classification of an unknown pattern is done according to the maximum voting , where each SVM votes for one class.

Summary

"Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani
"

Dimension Reduction

Curse of dimensionality

- Is the affliction caused by adding variables to models.
- Classification and regression models fail when too many variables are used.

How much data do we need?

- Vague.
- Depends on the complexity of the relationships between variables.
- Rule of thumb is 10 observations for every predictor.

What can we do?

- Incorporate domain knowledge to remove or combine categories.
- Detect information overlap between variables.
- **Create new set of variables combining the old variables.**

Principal Components Analysis

Goal: Reduce a set of numerical variables.

The idea: Remove the overlap of information between these variable. ["Information" is measured by the sum of the variances of the variables.]

Final product: A smaller number of numerical variables that contain most of the information.

Principal Components Analysis

How does PCA do this?

- Create new variables that are linear combinations of the original variables (i.e., they are weighted averages of the original variables).
- These linear combinations are uncorrelated (no information overlap), and only a few of them contain most of the original information.
- The new variables are called *principal components*.

Principal Components Analysis

- The first principal component of a set of features X_1, X_2, \dots, X_p is the normalized linear combination of the features

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$$

$$\sum_{j=1}^p \phi_{j1}^2 = 1$$

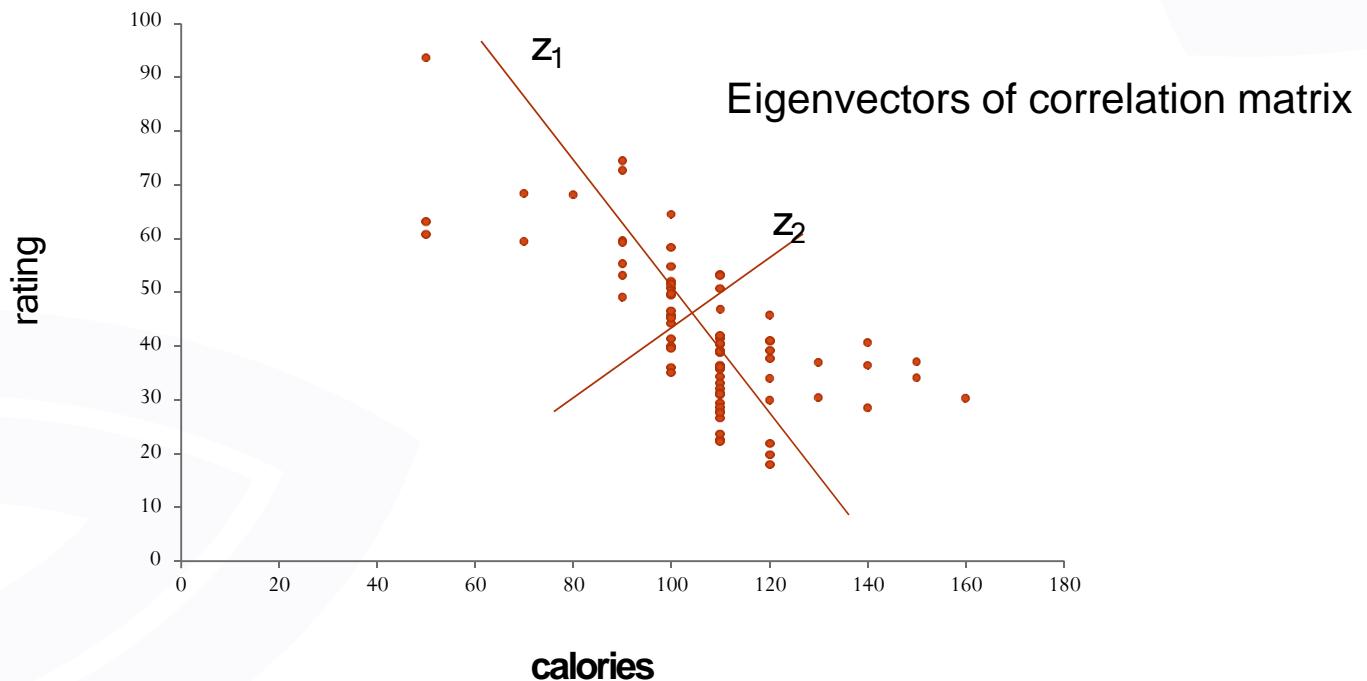
- $\phi_{11}, \dots, \phi_{p1}$ are the *loadings* of the first principal component

$$\underset{\phi_{11}, \dots, \phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^p \phi_{j1}^2 = 1.$$

First & Second Principal Components

Z_1 and Z_2 are two linear combinations.

- Z_1 has the highest variation (spread of values)
- Z_2 orthogonal

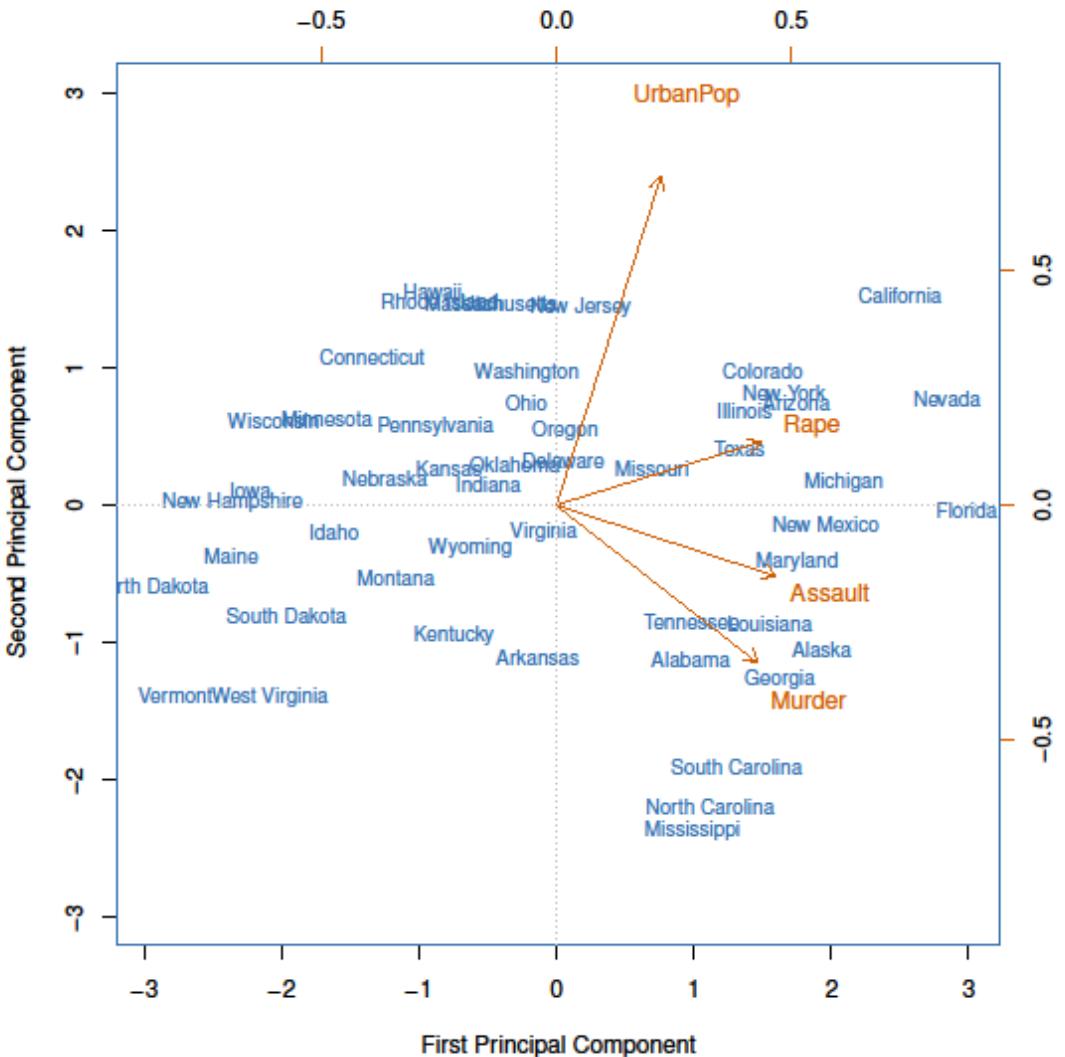


Example

- Loadings of the first two principal components for the USArrests dataset

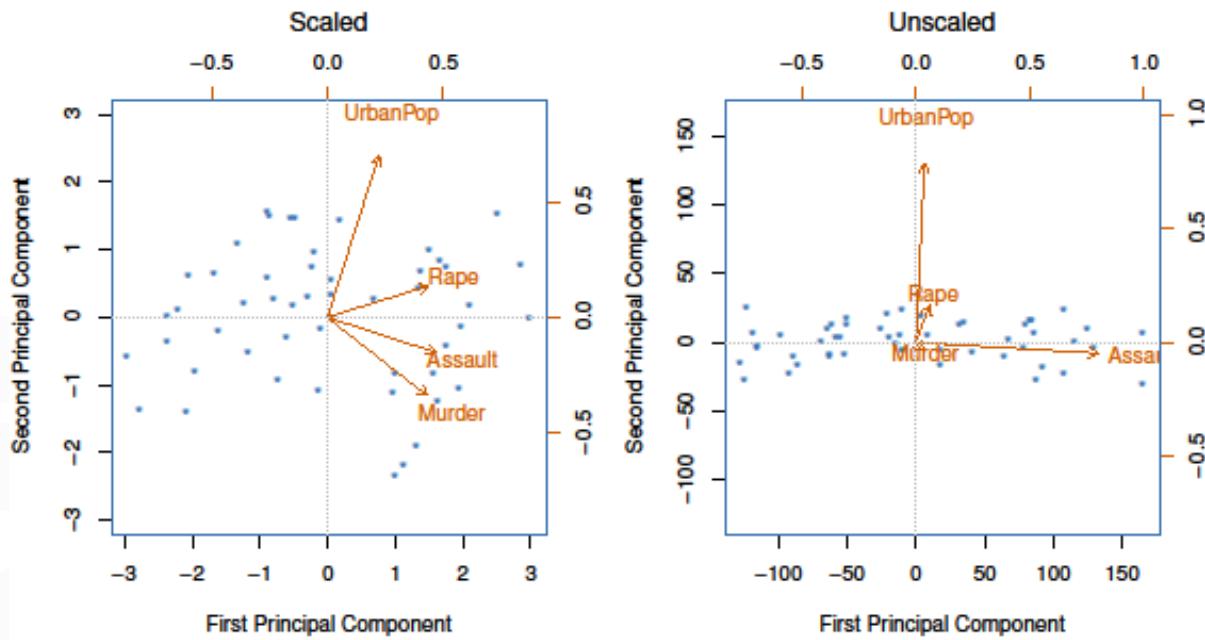
	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

Example



Scaling in PCA

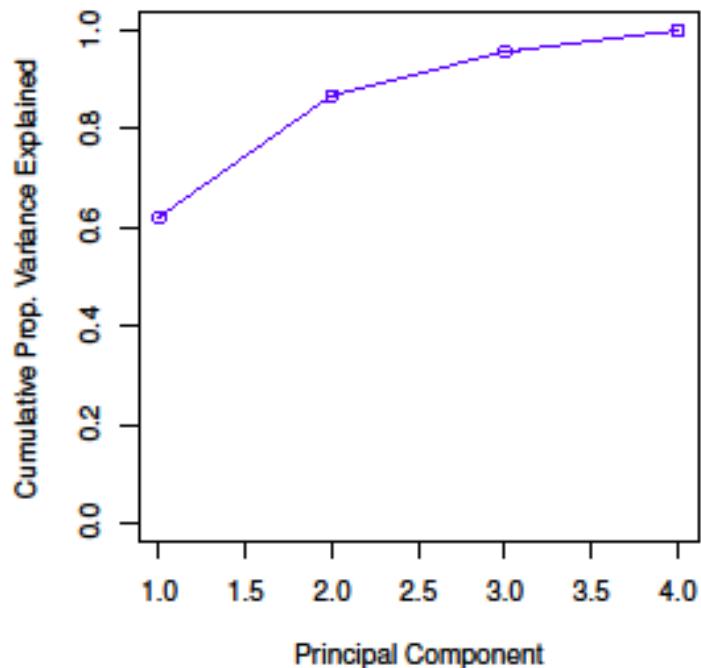
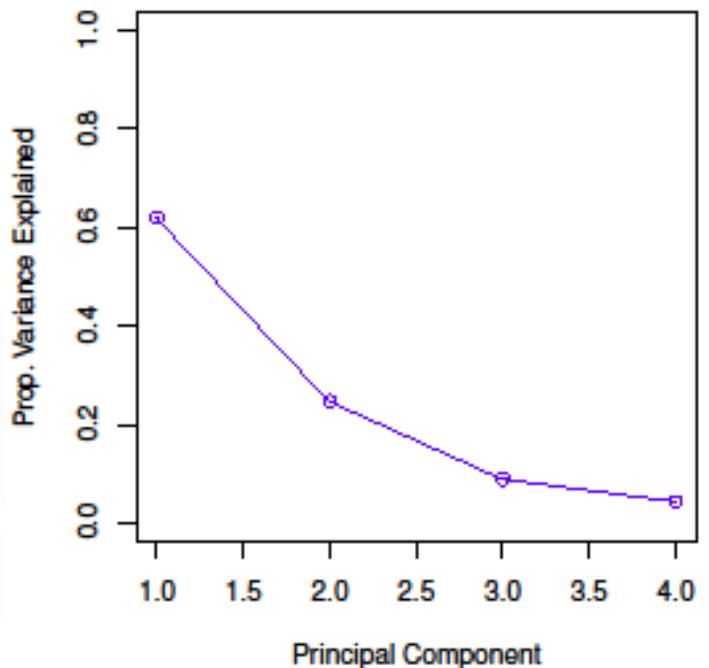
- The results obtained when we perform PCA will also depend on whether the variables have been individually scaled



The Proportion of Variance Explained

- We can calculate the total variance present in the dataset.
- We can also calculate the variance present in a given principal component.
- If we divide the variance present with the total variance, then, for each PC we can get a value which is called the variance explained.

The Proportion of Variance Explained



Generalization

- $X_1, X_2, X_3, \dots, X_p$, original p variables
- $Z_1, Z_2, Z_3, \dots, Z_p$, weighted averages of original variables
- All pairs of Z variables have 0 correlation
- Order Z 's by variance (z_1 largest, z_p smallest)
- Usually the first few Z variables contain most of the information, and so the rest can be dropped.

PCA in Classification/Prediction

- Apply PCA to training data.
- Decide how many PC's to use.
- Use variable weights in those PC's with validation/new data.
- This creates a new reduced set of predictors in validation/new data.
- Use the PCmatrix created on training data, applied to testing data

Example principal components

Cereal Name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins
100% Bran	N	C	70	4	1	130	10	5	6	280	25
100% Natural Bran	Q	C	120	3	5	15	2	8	8	135	0
All-Bran	K	C	70	4	1	260	9	7	5	320	25
All-Bran with Extra Fiber	K	C	50	4	0	140	14	8	0	330	25
Almond Delight	R	C	110	2	2	200	1	14	8		25
Apple Cinnamon Cheerios	G	C	110	2	2	180	1.5	10.5	10	70	25
Apple Jacks	K	C	110	2	0	125	1	11	14	30	25
Basic 4	G	C	130	3	2	210	2	18	8	100	25
Bran Chex	R	C	90	2	1	200	4	15	6	125	25
Bran Flakes	P	C	90	3	0	210	5	13	5	190	25
Cap'n'Crunch	Q	C	120	1	2	220	0	12	12	35	25
Cheerios	G	C	110	6	2	290	2	17	1	105	25
Cinnamon Toast Crunch	G	C	120	1	3	210	0	13	9	45	25
Clusters	G	C	110	3	2	140	2	13	7	105	25
Cocoa Puffs	G	C	110	1	1	180	0	12	13	55	25
Corn Chex	R	C	110	2	0	280	0	22	3	25	25
Corn Flakes	K	C	100	2	0	290	1	21	2	35	25
Corn Pops	K	C	110	1	0	90	1	13	12	20	25
Count Chocula	G	C	110	1	1	180	0	12	13	65	25
Cracklin' Oat Bran	K	C	110	3	3	140	4	10	7	160	25

Variable	Description
mfr	Manufacturer of cereal (American Home Food Products, General Mills, Kellogg, etc.)
type	Cold or hot
calories	Calories per serving
protein	Grams of protein
fat	Grams of fat
sodium	Milligrams of sodium
fiber	Grams of dietary fiber
carbo	Grams of complex carbohydrates
sugars	Grams of sugars
potass	Milligrams of potassium
vitamins	Vitamins and minerals: 0, 25, or 100, indicating the typical percentage of FDA recommended
shelf	Display shelf (1, 2, or 3, counting from the floor)
weight	Weight in ounces of one serving
cups	Number of cups in one serving
rating	Rating of the cereal calculated by consumer reports

Example (R output)

```
> pcs <- prcomp(na.omit(cereals.df[,-c(1:3)]))
> summary(pcs)

Importance of components:
              PC1       PC2       PC3       PC4       PC5       PC6       PC7
Standard deviation   83.7641  70.9143  22.64375 19.18148  8.42323  2.09167 1.69942
Proportion of Variance  0.5395  0.3867  0.03943  0.02829  0.00546  0.00034 0.00022
Cumulative Proportion  0.5395  0.9262  0.96560  0.99389  0.99935  0.99968 0.99991
                                         PC8       PC9       PC10      PC11      PC12      PC13
Standard deviation   0.77963  0.65783  0.37043  0.1864  0.06302 5.334e-08
Proportion of Variance 0.00005  0.00003  0.00001  0.0000  0.00000 0.000e+00
Cumulative Proportion 0.99995 0.99999 1.00000 1.0000 1.00000 1.000e+00

> pcs$rot[,1:5]

          PC1       PC2       PC3       PC4       PC5
calories  0.0779841812  0.0093115874 -0.6292057595 -0.6010214629  0.454958508
protein   -0.0007567806 -0.0088010282 -0.0010261160  0.0031999095  0.056175970
fat        -0.0001017834 -0.0026991522 -0.0161957859 -0.0252622140 -0.016098458
sodium    0.9802145422 -0.1408957901  0.1359018583 -0.0009680741  0.013948118
fiber     -0.0054127550 -0.0306807512  0.0181910456  0.0204721894  0.013605026
carbo     0.0172462607  0.0167832981 -0.0173699816  0.0259482087  0.349266966
sugars    0.0029888631  0.0002534853 -0.0977049979 -0.1154809105 -0.299066459
potass   -0.1349000039 -0.9865619808 -0.0367824989 -0.0421757390 -0.047150529
vitamins  0.0942933187 -0.0167288404 -0.6919777623  0.7141179984 -0.037008623
shelf    -0.0015414195 -0.0043603994 -0.0124888415  0.0056471836 -0.007876459
weight    0.0005120017 -0.0009992138 -0.0038059565 -0.0025464145  0.003022113
cups      0.0005101111  0.0015910125 -0.0006943214  0.0009853800  0.002148458
rating   -0.0752962922 -0.0717421528  0.3079471212  0.3345338994  0.757708025
```

Summary

"Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani
"

Clustering: The Main Idea

Goal: Form groups (clusters) of similar records

Used for **segmenting markets** into groups of similar customers

Example: Claritas segmented US neighborhoods based on demographics & income: "Furs & station wagons," "Money & Brains", ...

Other Applications

- Periodic table of the elements
- Classification of species
- Grouping securities in portfolios
- Grouping firms for structural analysis of economy
- Army uniform sizes

Example: Public Utilities

Goal: find clusters of similar utilities

Data: 22 firms, 8 variables

Fixed-charge covering ratio

Rate of return on capital

Cost per kilowatt capacity

Annual load factor

Growth in peak demand

Sales

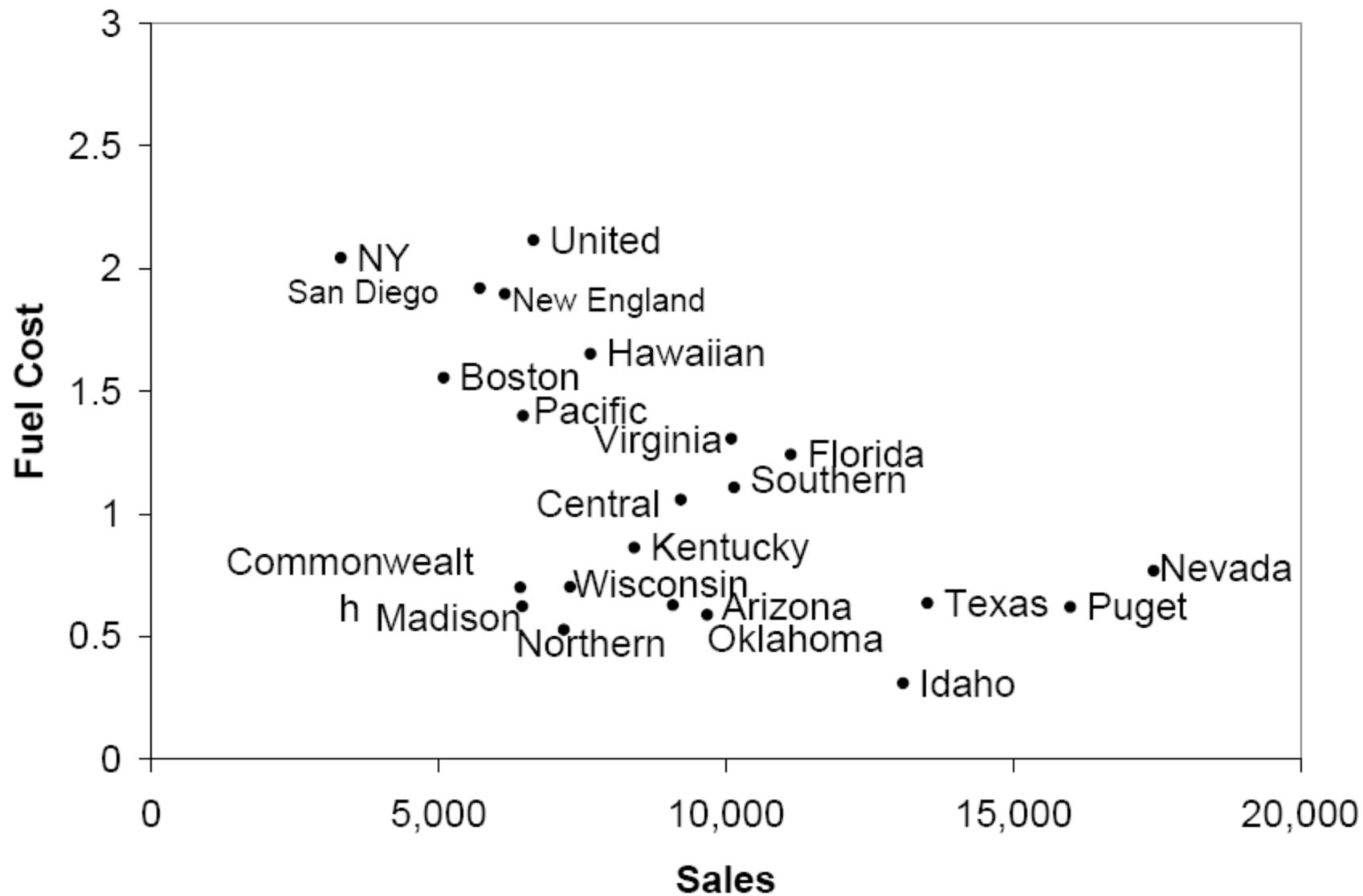
% nuclear

Fuel costs per kwh

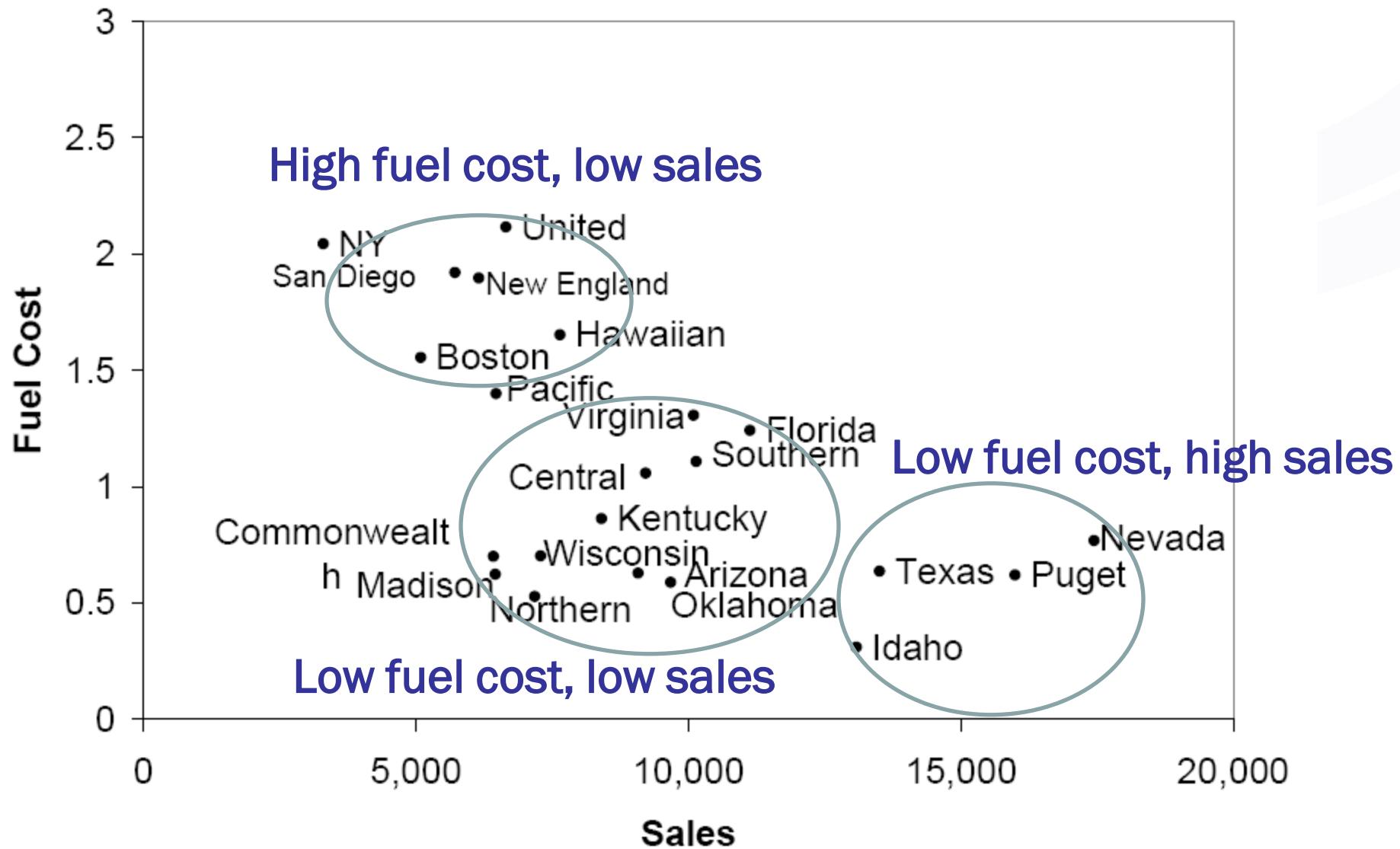
Example: Public Utilities

Company	Fixed_charge	RoR	Cost	Load	Δ Demand	Sales	Nuclear	Fuel_Cost	
Arizona	1.06	9.2	151	54.4		1.6	9077	0	0.628
Boston	0.89	10.3	202	57.9		2.2	5088	25.3	1.555
Central	1.43	15.4	113	53		3.4	9212	0	1.058
Commonwealth	1.02	11.2	168	56		0.3	6423	34.3	0.7
Con Ed NY	1.49	8.8	192	51.2		1	3300	15.6	2.044
Florida	1.32	13.5	111	60		-2.2	11127	22.5	1.241
Hawaiian	1.22	12.2	175	67.6		2.2	7642	0	1.652
Idaho	1.1	9.2	245	57		3.3	13082	0	0.309
Kentucky	1.34	13	168	60.4		7.2	8406	0	0.862
Madison	1.12	12.4	197	53		2.7	6455	39.2	0.623
Nevada	0.75	7.5	173	51.5		6.5	17441	0	0.768
New England	1.13	10.9	178	62		3.7	6154	0	1.897
Northern	1.15	12.7	199	53.7		6.4	7179	50.2	0.527
Oklahoma	1.09	12	96	49.8		1.4	9673	0	0.588
Pacific	0.96	7.6	164	62.2		-0.1	6468	0.9	1.4
Puget	1.16	9.9	252	56		9.2	15991	0	0.62
San Diego	0.76	6.4	136	61.9		9	5714	8.3	1.92
Southern	1.05	12.6	150	56.7		2.7	10140	0	1.108
Texas	1.16	11.7	104	54		-2.1	13507	0	0.636
Wisconsin	1.2	11.8	148	59.9		3.5	7287	41.1	0.702
United	1.04	8.6	204	61		3.5	6650	0	2.116
Virginia	1.07	9.3	174	54.3		5.9	10093	26.6	1.306

Sales & Fuel Cost



Sales & Fuel Cost: rough clusters are seen



Extension to More Than 2 Dimensions

In prior example, clustering was done by eye

Multiple dimensions require formal algorithm
with

- A **distance measure**
- A way to use the distance measure in forming clusters

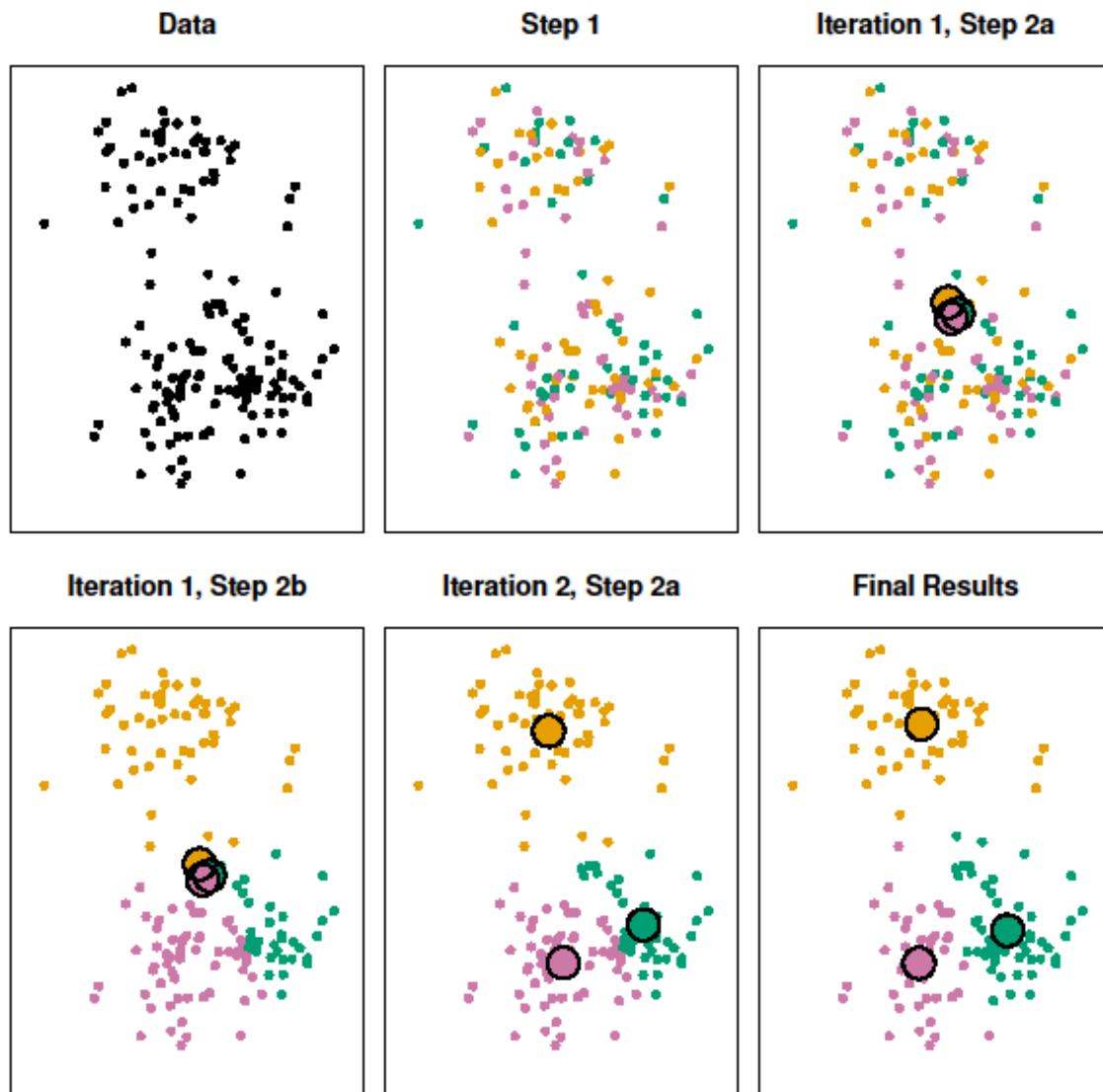
We will consider two types of algorithms:
hierarchical and **non-hierarchical**

Nonhierarchical Clustering: K-Means Clustering

K-Means Clustering Algorithm

1. Choose # of clusters desired, k
2. Start with a partition into k clusters
Often based on random selection of k centroids
3. At each step, move each record to cluster with closest centroid
4. Recompute centroids, repeat step 3
5. Stop when moving records increases within-cluster dispersion

K-Means Clustering Algorithm



K-means Algorithm: Choosing k and Initial Partitioning

Choose k based on the how results will be used
e.g., “How many market segments do we want?”

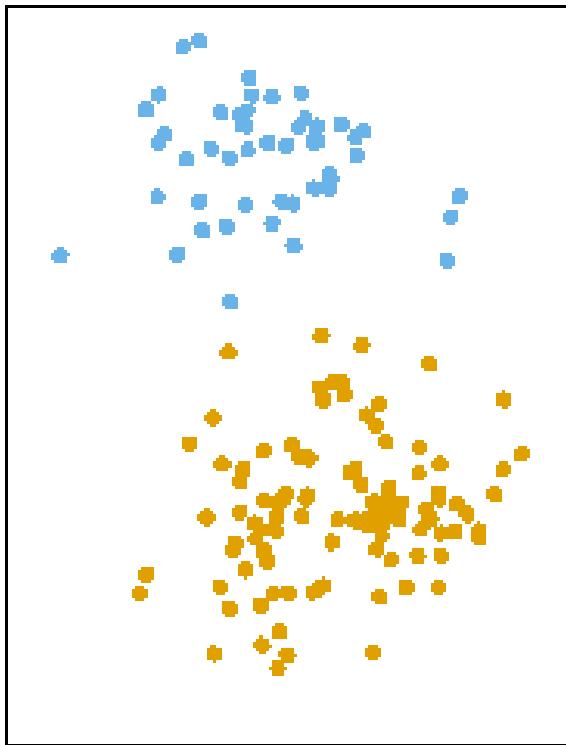
Also experiment with slightly different k 's

Initial partition into clusters can be random, or
based on domain knowledge

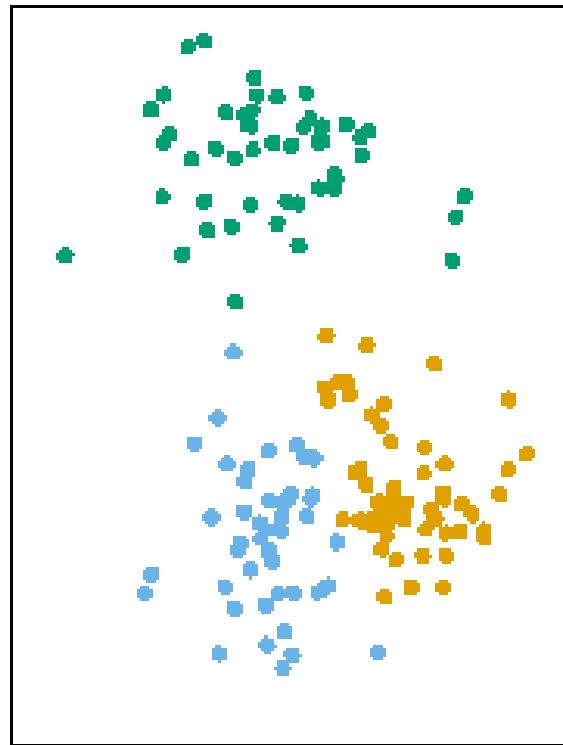
If random partition, repeat the process with different
random partitions

K-means Algorithm: Choosing k

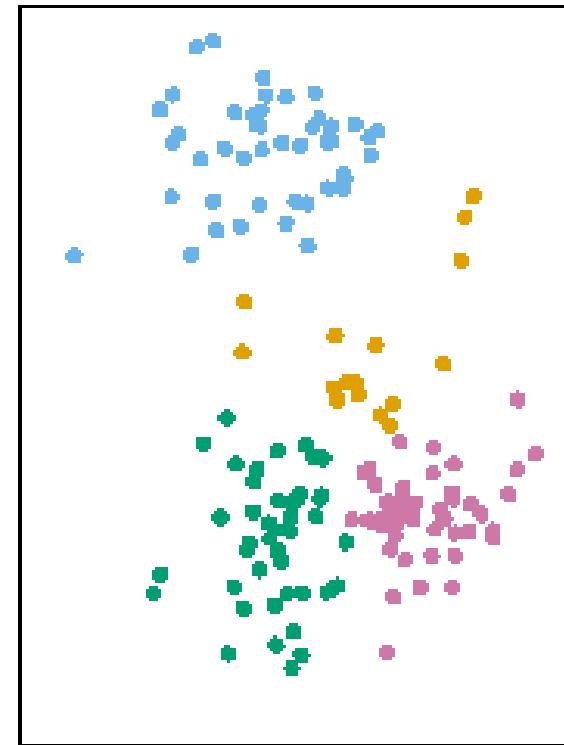
K=2



K=3



K=4



XLMiner Output: Cluster Centroids

Cluster	Fixed_charge	RoR	Cost	Load_factor
Cluster-1	0.89	10.3	202	57.9
Cluster-2	1.43	15.4	113	53
Cluster-3	1.06	9.2	151	54.4

We chose $k = 3$

4 of the 8 variables are shown

Distance Between Clusters

Distance between	Cluster-1	Cluster-2	Cluster-3
Cluster-1	0	5.03216253	3.16901457
Cluster-2	5.03216253	0	3.76581196
Cluster-3	3.16901457	3.76581196	0

Clusters 1 and 2 are relatively well-separated from each other, while cluster 3 not as much

Within-Cluster Dispersion

Data summary (In Original coordinates)

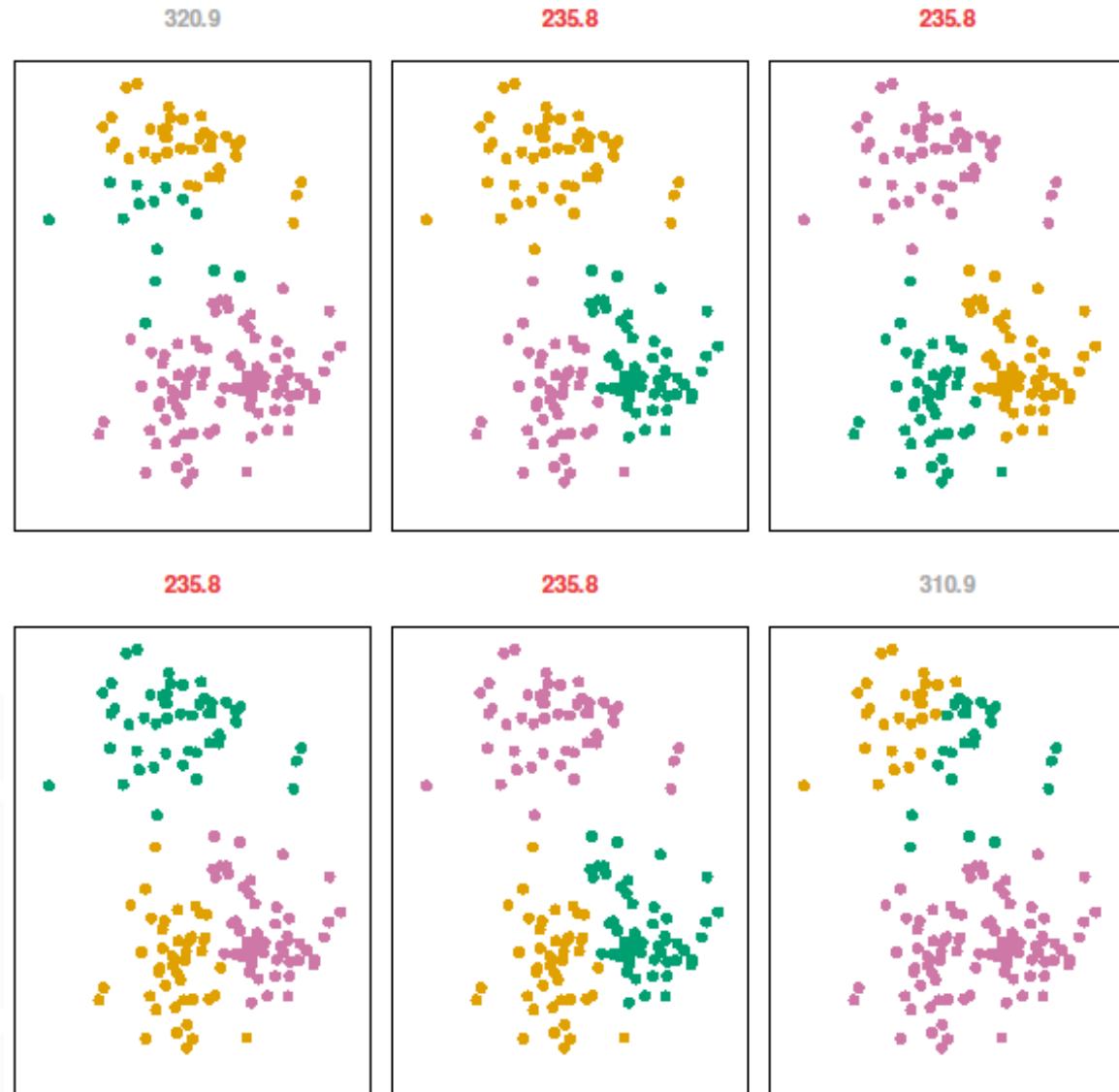
Cluster	#Obs	Average distance in cluster
Cluster-1	12	1748.348058
Cluster-2	3	907.6919822
Cluster-3	7	3625.242085
Overall	22	2230.906692

Clusters 1 and 2 are relatively tight, cluster 3 very loose

Conclusion: Clusters 1 & 2 well defined, not so for cluster 3

Next step: try again with $k=2$ or $k=4$

Within-Cluster Dispersion



Hierarchical Clustering

Hierarchical Methods

Agglomerative Methods

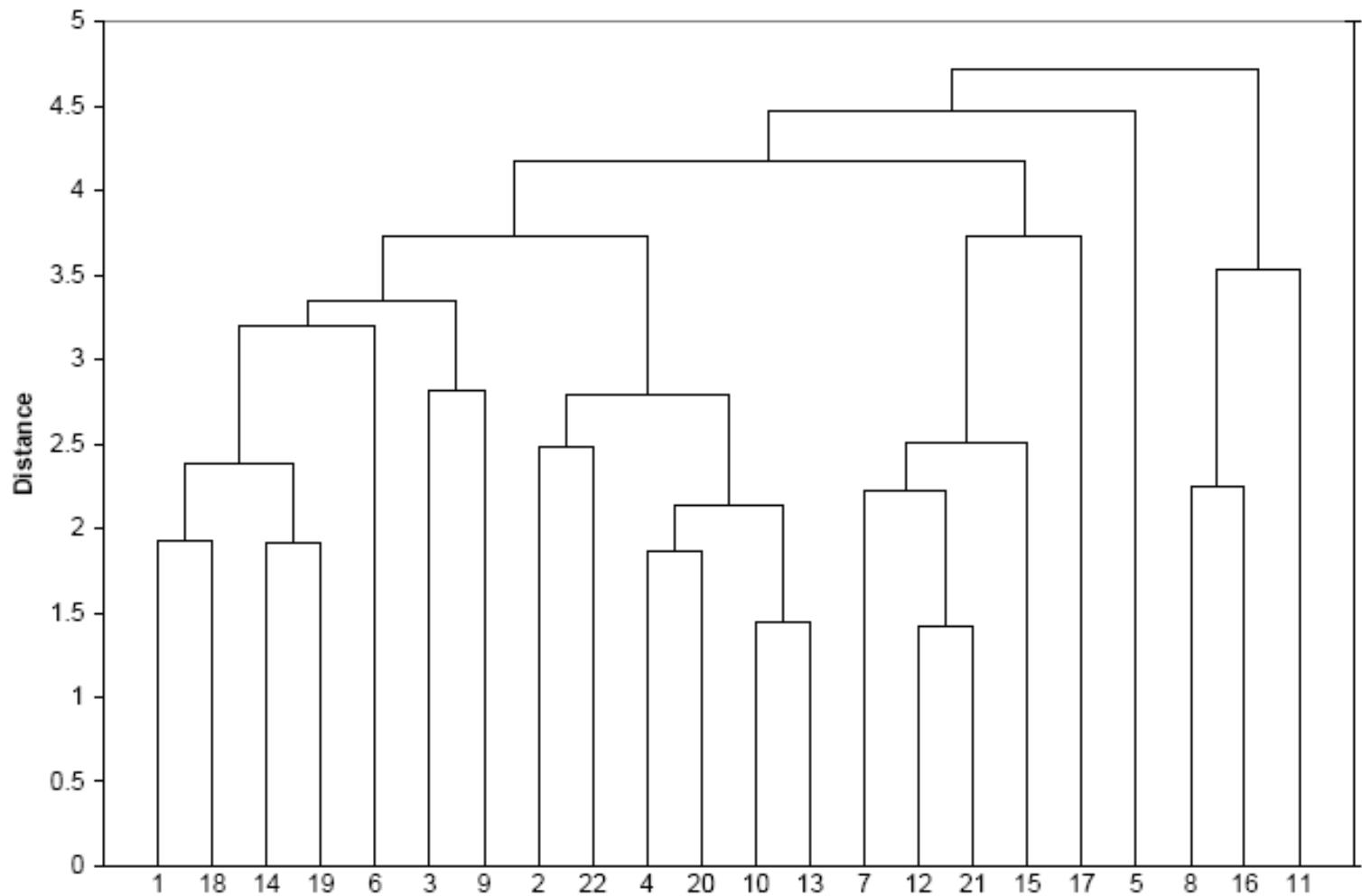
- Begin with n-clusters (each record its own cluster)
- Keep joining records into clusters until one cluster is left (the entire data set)
- Most popular

Divisive Methods

- Start with one all-inclusive cluster
- Repeatedly divide into smaller clusters

A Dendrogram shows the cluster hierarchy

Dendrogram(Average linkage)



Measuring Distance

Between records

Between clusters

Measuring Distance Between Records

Distance Between Two Records

Euclidean Distance is most popular:

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2}$$

Normalizing

Problem: Raw distance measures are highly influenced by scale of measurements

Solution: normalize (standardize) the data first

- Subtract mean, divide by std. deviation
- Also called **z-scores**

Example: Normalization

For 22 utilities:

Avg. sales = 8,914

Std. dev. = 3,550

Normalized score for Arizona sales:

$$(9,077 - 8,914) / 3,550 = 0.046$$

Example: Normalization

Company	Sales	Fuel Cost	NormSales	NormFuel
Arizona Public Service	9077	0.628	0.0459	-0.8537
Boston Edison Co.	5088	1.555	-1.0778	0.8133
Central Louisiana Co.	9212	1.058	0.0839	-0.0804
Commonwealth Edison Co.	6423	0.7	-0.7017	-0.7242
Consolidated Edison Co. (NY)	3300	2.044	-1.5814	1.6926
Florida Power & Light Co.	11127	1.241	0.6234	0.2486
Hawaiian Electric Co.	7642	1.652	-0.3583	0.9877
Idaho Power Co.	13082	0.309	1.1741	-1.4273
Kentucky Utilities Co.	8406	0.862	-0.1431	-0.4329
Madison Gas & Electric Co.	6455	0.623	-0.6927	-0.8627
Nevada Power Co.	17441	0.768	2.4020	-0.6019
New England Electric Co.	6154	1.897	-0.7775	1.4283
Northern States Power Co.	7179	0.527	-0.4887	-1.0353
Oklahoma Gas & Electric Co.	9673	0.588	0.2138	-0.9256
Pacific Gas & Electric Co.	6468	1.4	-0.6890	0.5346
Puget Sound Power & Light Co.	15991	0.62	1.9935	-0.8681
San Diego Gas & Electric Co.	5714	1.92	-0.9014	1.4697
The Southern Co.	10140	1.108	0.3453	0.0095
Texas Utilities Co.	13507	0.636	1.2938	-0.8393
Wisconsin Electric Power Co.	7287	0.702	-0.4583	-0.7206
United Illuminating Co.	6650	2.116	-0.6378	1.8221
Virginia Electric & Power Co.	10093	1.306	0.3321	0.3655
Mean	8914.05	1.10	0.00	0.00
Standard deviation	3549.98	0.56	1.00	1.00

For Categorical Data: Similarity

To measure the distance between records in terms of two 0/1 variables, create table with counts:

	0	1
0	a	b
1	c	d

Similarity metrics based on this table:

- Matching coef. = $(a+d)/p$
- Jaquard's coef. = $d/(b+c+d)$
 - Use in cases where a matching “1” is much greater evidence of similarity than matching “0” (e.g. “owns Corvette”)

Other Distance Measures

- Correlation-based similarity
- Statistical distance (Mahalanobis)
- Manhattan distance (absolute differences)
- Maximum coordinate distance
- Gower's similarity (for mixed variable types:
continuous & categorical)

Measuring Distance Between Clusters

Minimum Distance (Cluster A to Cluster B)

- Also called **single linkage**
- Distance between two clusters is the distance between the pair of records A_i and B_j that are closest

Maximum Distance (Cluster A to Cluster B)

- Also called **complete linkage**
- Distance between two clusters is the distance between the pair of records A_i and B_j that are farthest from each other

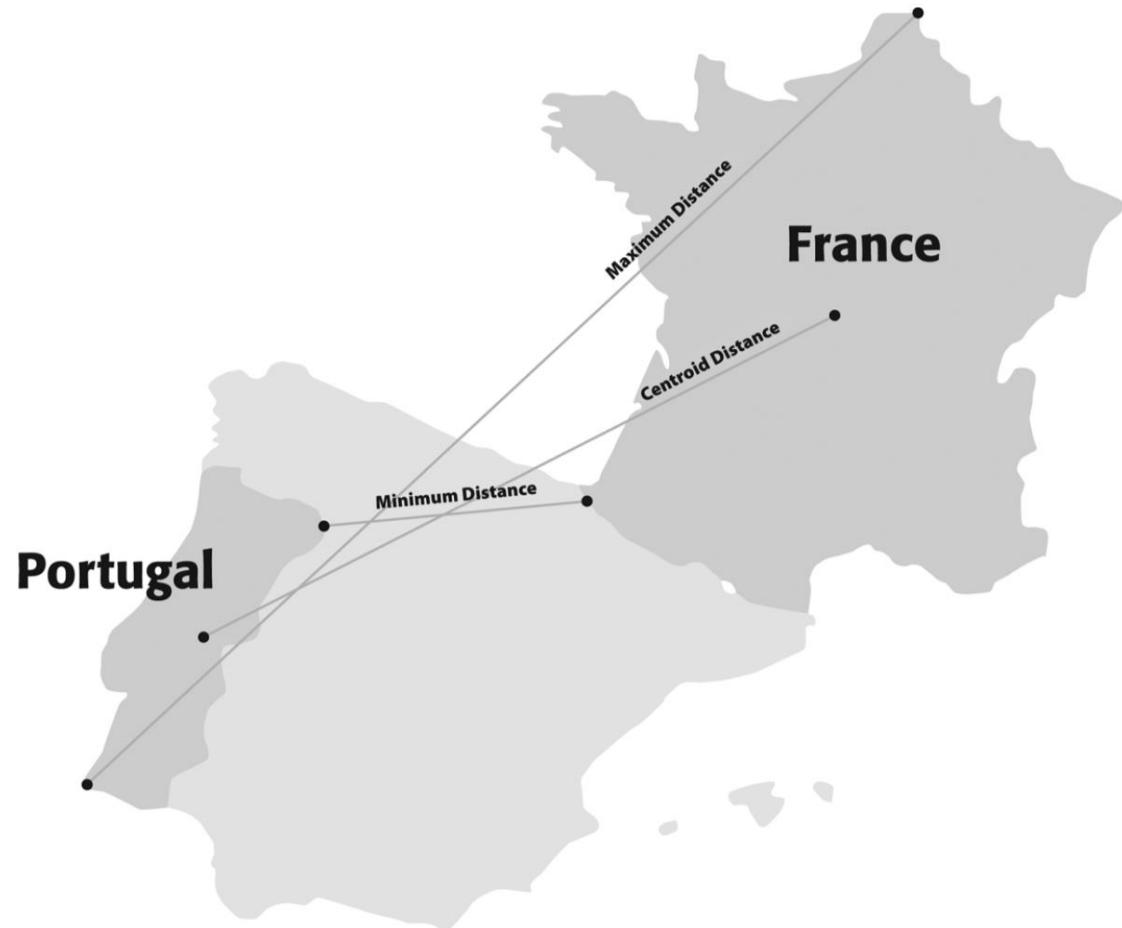
Average Distance

- Also called **average linkage**
- Distance between two clusters is the average of all possible pair-wise distances

Centroid Distance

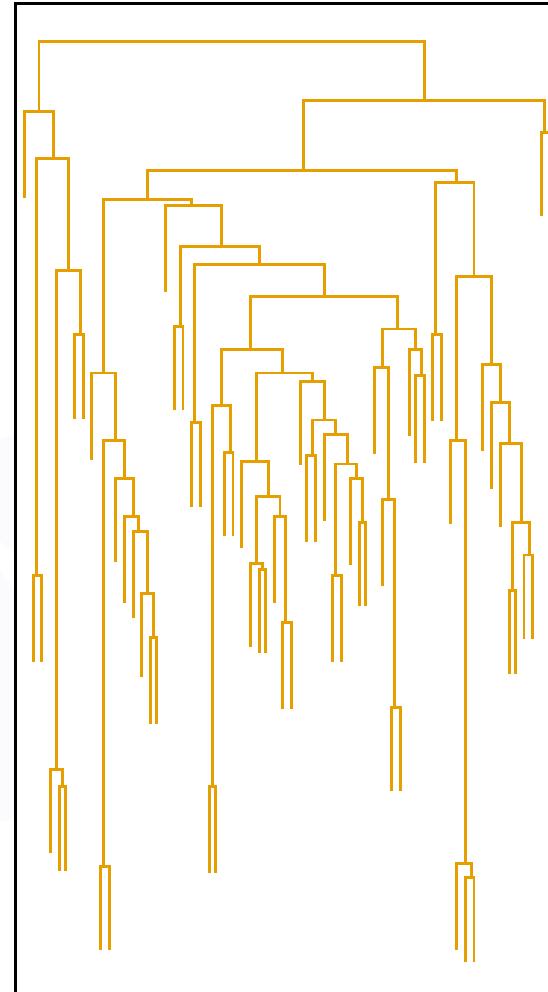
- Distance between two clusters is the distance between the two cluster centroids.
- Centroid is the vector of variable averages for all records in a cluster

Distance

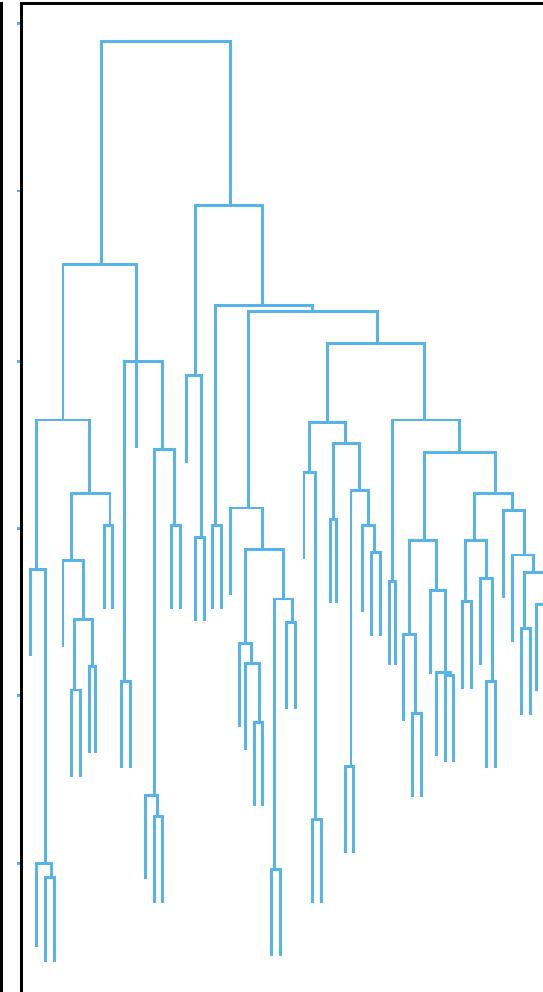


Distance

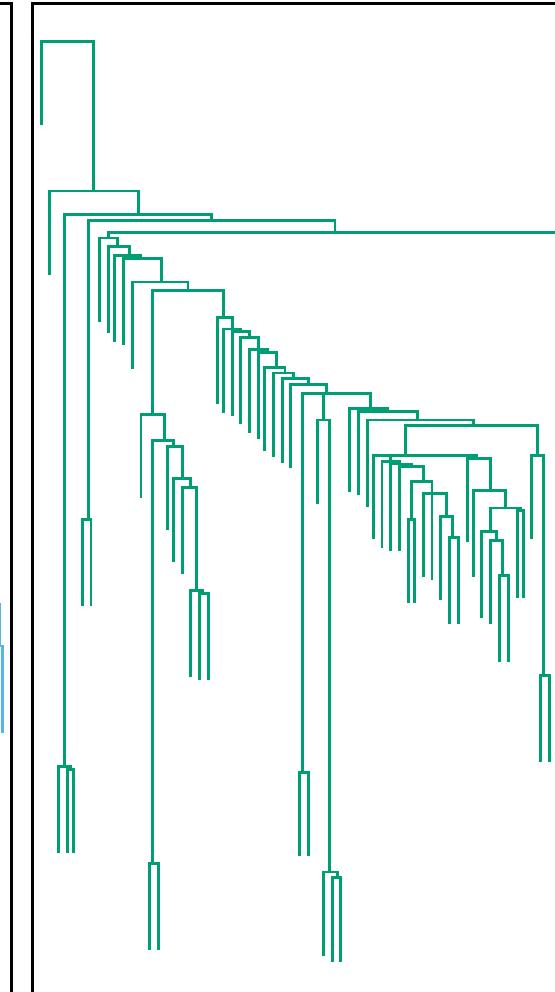
Average Linkage



Complete Linkage



Single Linkage



The Hierarchical Clustering Steps (Using Agglomerative Method)

1. Start with n clusters (each record is its own cluster)
2. Merge two closest records into one cluster
3. At each successive step, the two clusters closest to each other are merged

Dendrogram, from bottom up, illustrates the process

Example clustering

TABLE 15.3 DISTANCE MATRIX BETWEEN PAIRS OF THE FIRST FIVE UTILITIES, USING EUCLIDEAN DISTANCE AND NORMALIZED MEASUREMENTS

	Arizona	Boston	Central	Commonwealth	Consolidated
Arizona	0				
Boston	2.01	0			
Central	0.77	1.47	0		
Commonwealth	0.76	1.58	1.02	0	
Consolidated	3.02	1.01	2.43	2.57	0

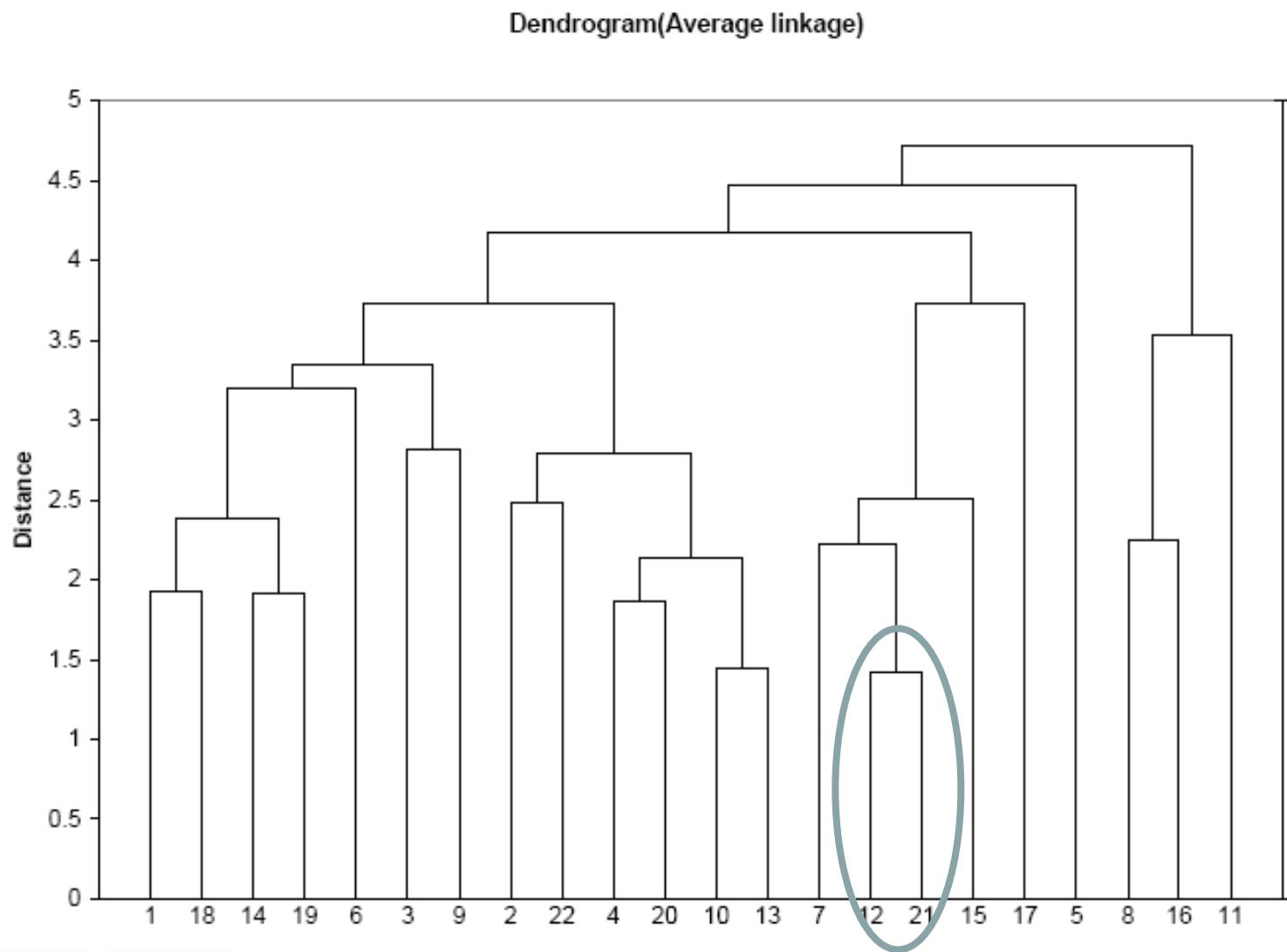
Example clustering

	Arizona	Boston	Central	Commonwealth	Consolidated
Arizona	0				
Boston	2.01	0			
Central	0.77	1.47	0		
Commonwealth	0.76	1.58	1.02	0	
Consolidated	3.02	1.01	2.43	2.57	0



	Arizona–Commonwealth	Boston	Central	Consolidated
Arizona–Commonwealth	0			
Boston	$\min(2.01, 1.58)$	0		
Central	$\min(0.77, 1.02)$	1.47	0	
Consolidated	$\min(3.02, 2.57)$	1.01	2.43	0

Records 12 & 21 form first cluster



Reading the Dendrogram

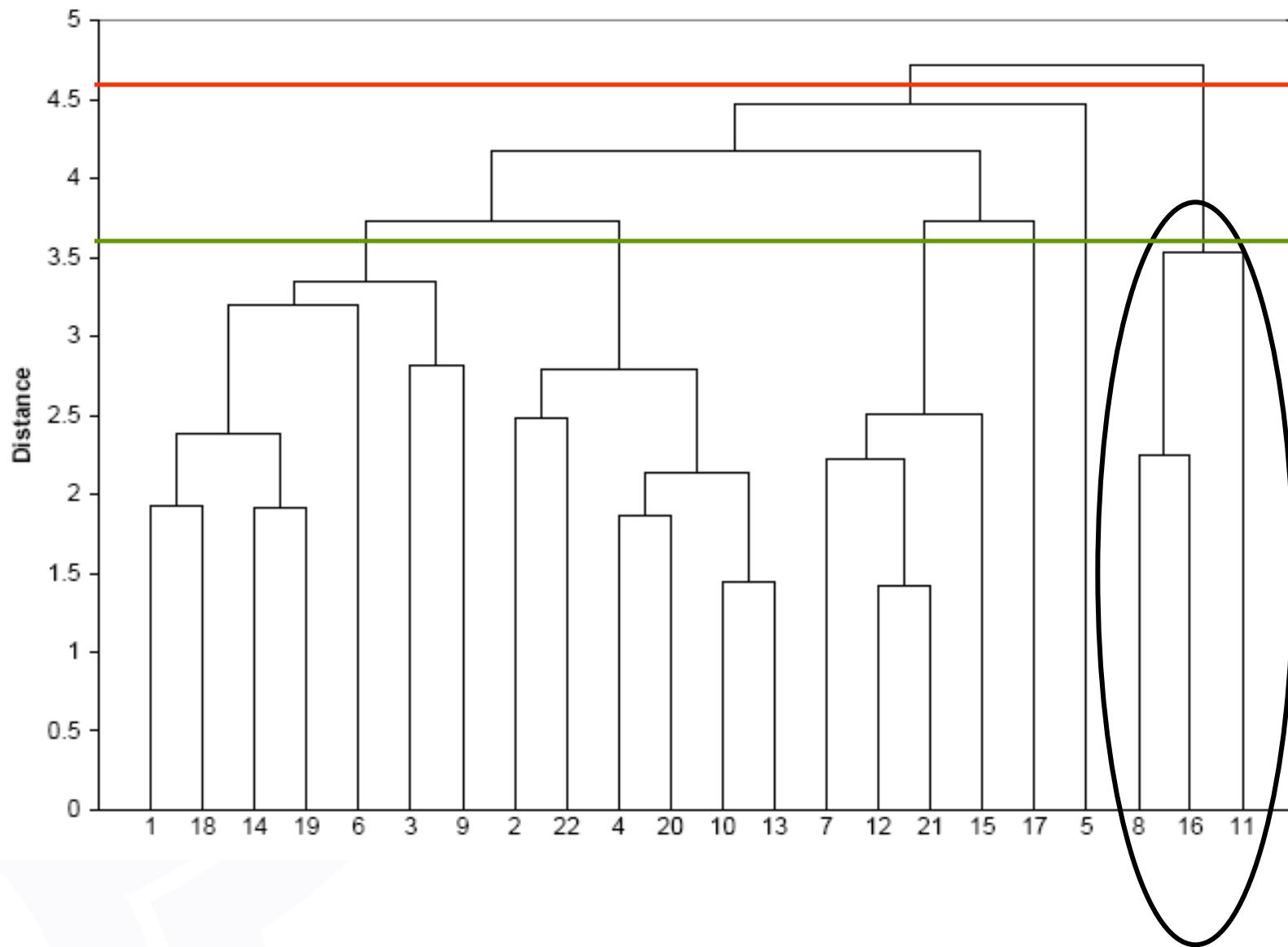
See process of clustering: Lines connected lower down are merged earlier

- 10 and 13 will be merged next, after 12 & 21

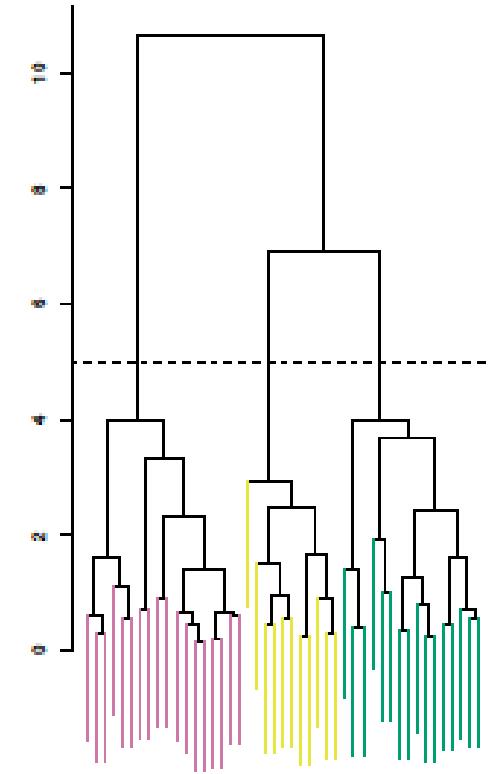
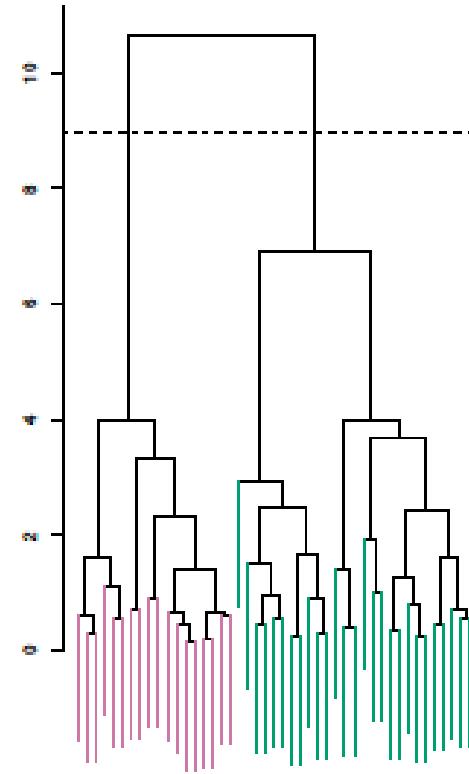
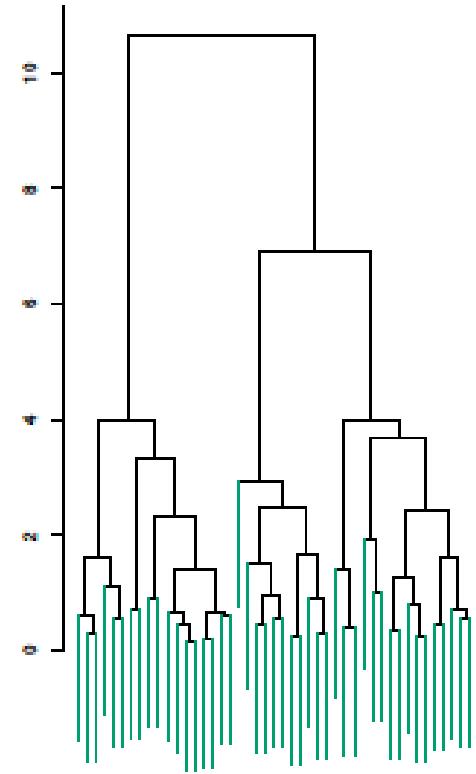
Determining number of clusters: For a given “distance between clusters”, a horizontal line intersects the clusters that are that far apart, to create clusters

- E.g., at distance of 4.6 (**red line** in next slide), data can be reduced to 2 clusters -- The smaller of the two is circled
- At distance of 3.6 (**green line**) data can be reduced to 6 clusters, including the circled cluster

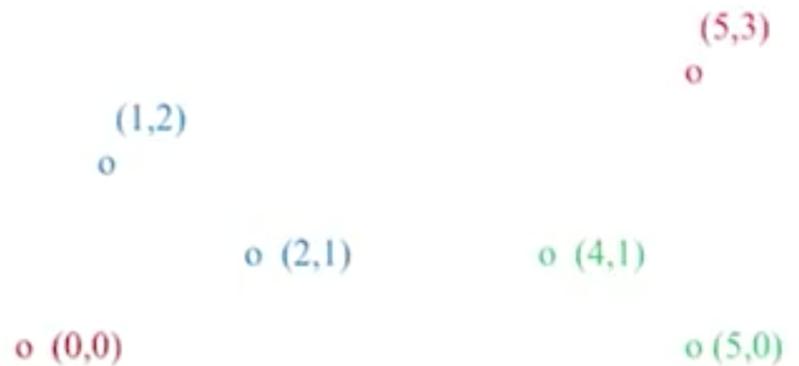
Dendrogram(Average linkage)



Reading the Dendrogram



Reading the Dendrogram

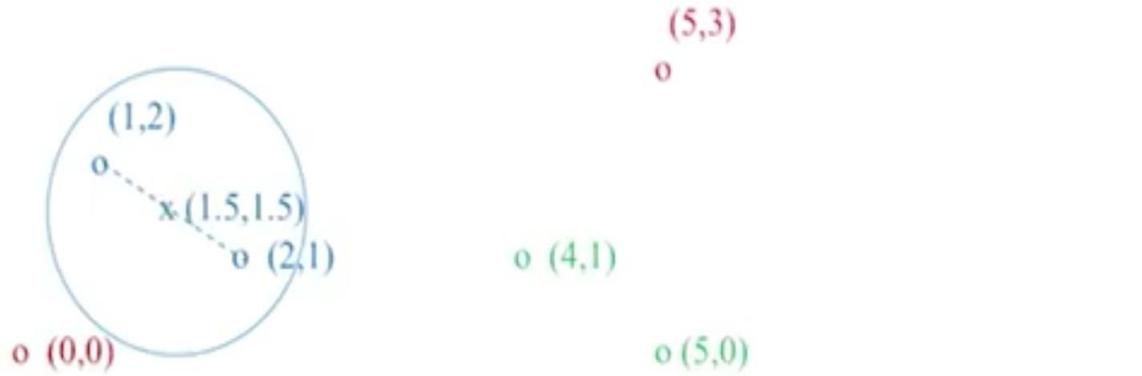
**Data:**

o ... data point

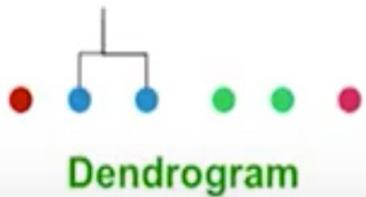
x ... centroid

**Dendrogram**

Reading the Dendrogram

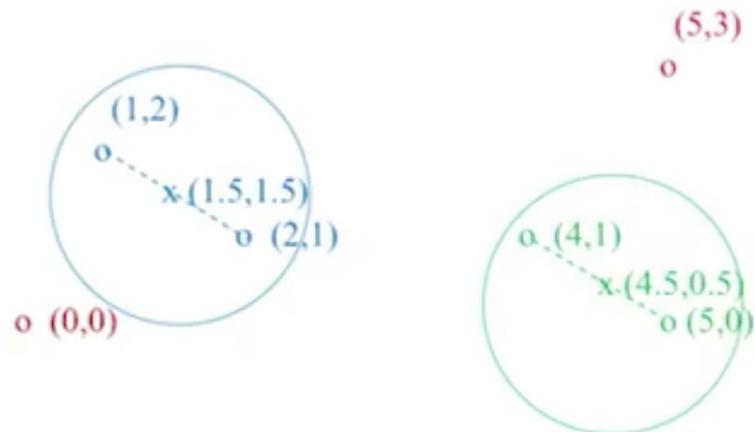


Data:
o ... data point
x ... centroid

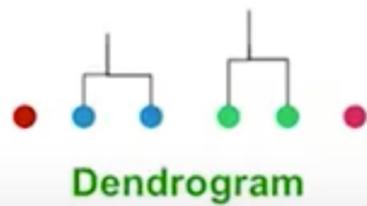


Dendrogram

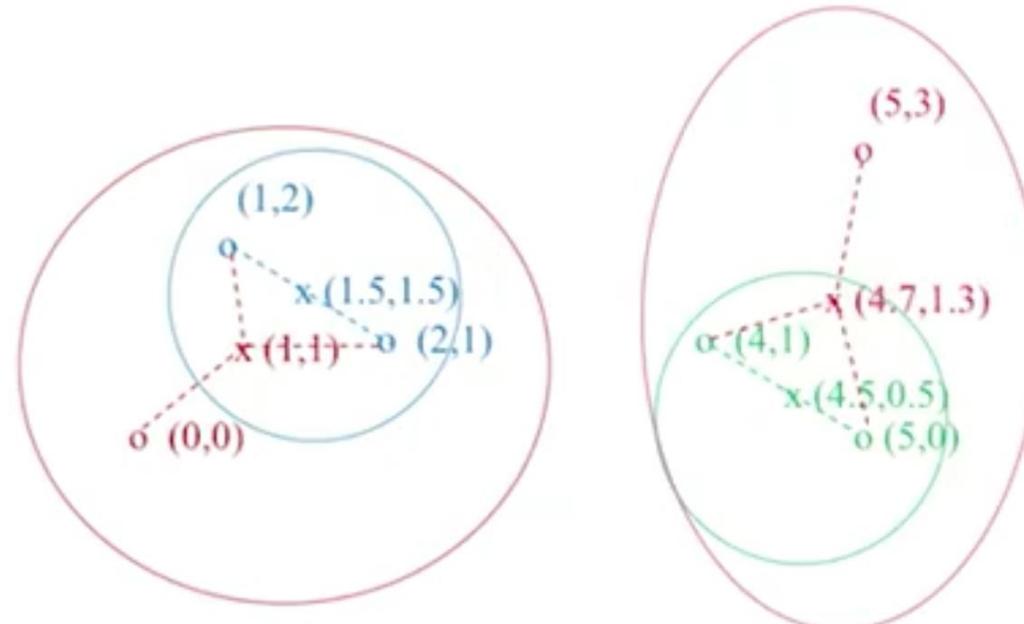
Reading the Dendrogram



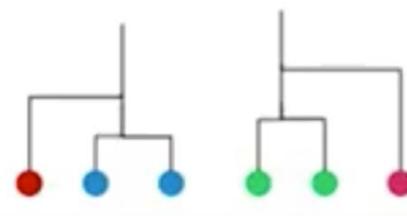
Data:
o ... data point
x ... centroid



Reading the Dendrogram



Data:
o ... data point
x ... centroid



Dendrogram

Validating Clusters

Interpretation

Goal: obtain meaningful and useful clusters

Caveats:

- (1) Random chance can often produce apparent clusters
- (2) Different cluster methods produce different results

Solutions:

- Obtain summary statistics
- Also review clusters in terms of variables **not** used in clustering
- **Label the cluster** (e.g. clustering of financial firms in 2008 might yield label like “midsize, sub-prime loser”)

Desirable Cluster Features

Stability – are clusters and cluster assignments sensitive to slight changes in inputs? Are cluster assignments in partition B similar to partition A?

Separation – check ratio of between-cluster variation to within-cluster variation (higher is better)

Summary

- Cluster analysis is an exploratory tool. Useful only when it produces **meaningful** clusters
- **Hierarchical** clustering gives visual representation of different levels of clustering
 - On other hand, due to non-iterative nature, it can be unstable, can vary highly depending on settings, and is computationally expensive
- **Non-hierarchical** is computationally cheap and more stable; requires user to set k
- Can use both methods
- Be wary of chance results; data may not have definitive “real” clusters

