

ST 443: Machine Learning and Data Mining

Dr. Xinghao Qiao

Columbia House, Room 5.15

X.Qiao@lse.ac.uk

Office Hours: Tuesday 4:30–5:30pm
Department of Statistics



TA: Cheng Chen, C.Chen44@lse.ac.uk, OH: TBD
TA: Yirui Liu, Y.Liu110@lse.ac.uk, OH: TBD

Lecture 1

Evaluation

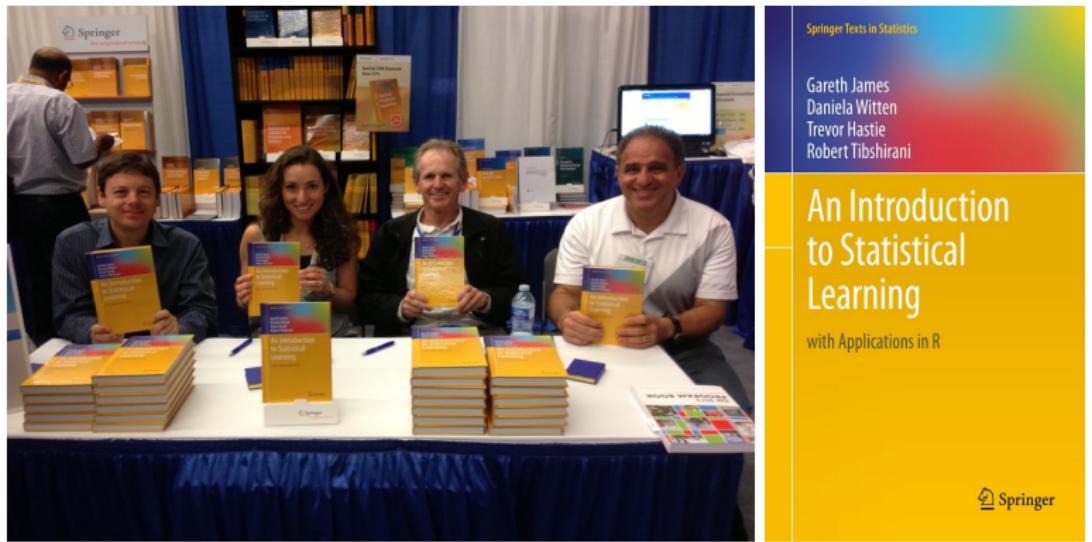
- The course grade will be made up from
 - ① **70%**: 2-hour exam (summer term)
 - ② **30%**: Group projects
 - ③ **0%**: 5 homework
- Group project
 - i : Similar to a take-home exam.
 - ii : Apply machine learning techniques on real data.
 - iii : Report (11th week).

Software

- There are many possible statistical software (or data mining) packages that one could use.
- They are generally very expensive.
- We will use **R** in class. The **Python** version will also be provided.
- It is very powerful (I use it in my own research).
- Best of all it is free.

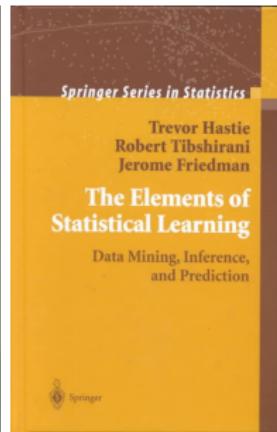
Course Materials

- Required textbook: “An Introduction to Statistical Learning with Applications in R” (ISLR).
- Free download: <http://www-bcf.usc.edu/gareth/ISL/>



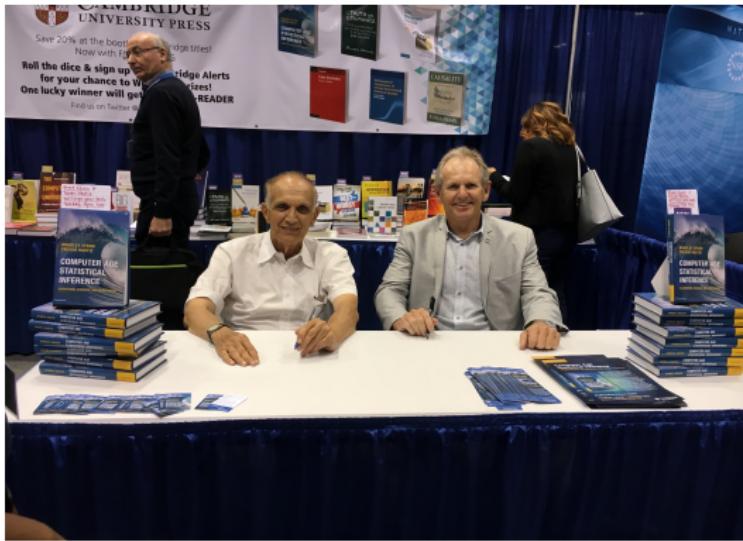
Course Materials

- Highly recommended book: “The Elements of Statistical Learning” (ESL).
- Free download: <http://statweb.stanford.edu/~tibs/ElemStatLearn/>



Course Materials

- Highly recommended book: “Computer Age Statistical Inference: Algorithms, Evidence and Data Science” .
- Free download: <http://web.stanford.edu/~hastie/CASI/>



Labs (Very Important!!!)

- This is intended to be a not-very-theoretical course.
 - We plan to spend 1/3 of the class sessions on labs.
 - For lab sessions, we will work through, in R, the statistical methods that we have learned about in lectures.

Outline

- ① Introduction to Statistical Machine Learning
- ② Review of Linear Regression
- ③ Classification
- ④ Resampling Methods
- ⑤ Linear Model Selection and Regularization
- ⑥ Non-linear Models
- ⑦ Tree-based Methods
- ⑧ Support Vector Machines
- ⑨ Unsupervised Learning

Neural network and deep learning will be covered in ST449

Machine learning from a Bayesian perspective will be covered in ST451.

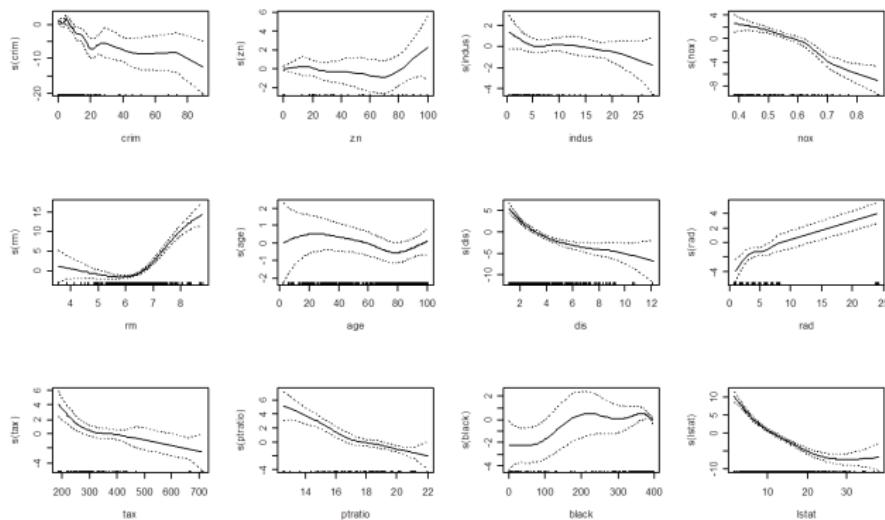
Some Interesting Data Sets

- Income classification
- Credit screening
- Insurance
- Predict housing price
- Response to direct mailings
- Predict a baseball player's salary
- Predict mpg for a car
- Predict heart decease
- Predict ZIP code
- Is an email a SPAM?
- ...

We may not have time to cover every data set in class. However, I have all the data sets available for you to practice with.

Boston Housing Data

- Aim to predict median value of homes based on crime rate, zoning, location to river, air quality, schools, etc.



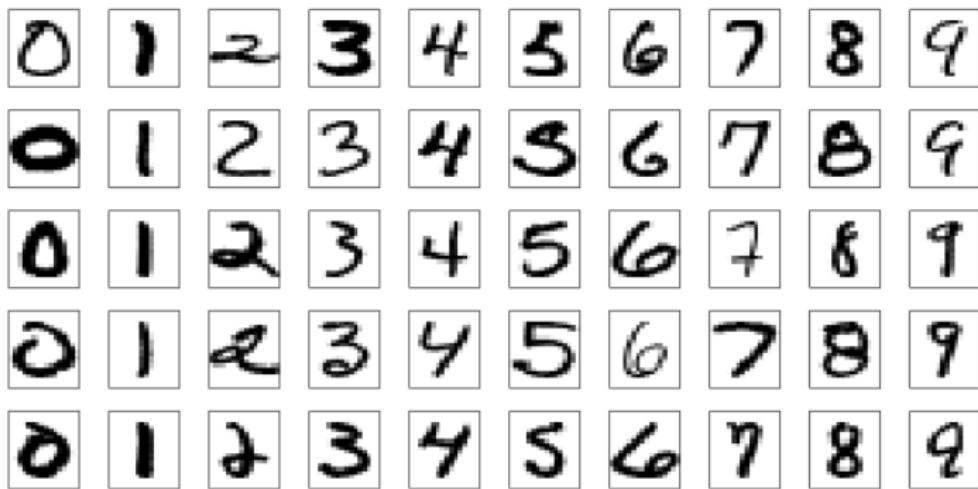
SPAM Detection

- Data consists of 4600 emails sent to an individual (named George, at HP labs at early years). Each is labeled spam or email.
- Features: relative frequencies of 57 of commonly occurring words and punctuation marks.
- Aim to build a customized spam filer.

	george	you	hp	free	!	edu	remove
spam	0.00	2.26	0.02	0.52	0.51	0.01	0.28
email	1.27	1.27	0.90	0.07	0.11	0.29	0.01

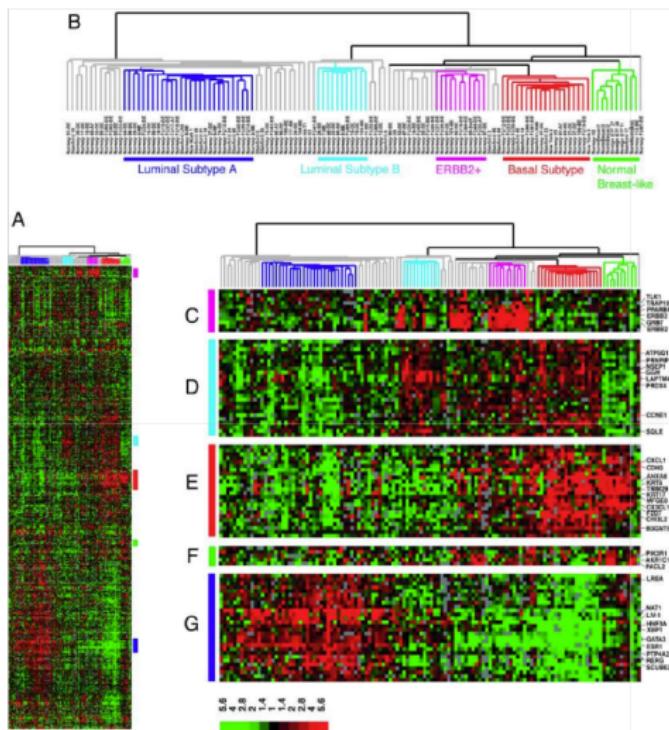
Zip Code Data

- Multiple hidden layer neural networks (deep learning) have been used to read handwritten zip codes on US Mail.
- This is a classification problem with 256 predictors (gray scale values on 16 by 16 grids) and 10 possible categories.



Genetics Data

- Aim to classify a tissue sample into one of several cancer classes based on a gene expression profile.



Big Data in Statistics

- What is big data from a statistical perspective?
- n : number of observations; p : number of features (dimensionality).

Big Data in Statistics

- What is big data from a statistical perspective?
 - n : number of observations; p : number of features (dimensionality).
- ① **Big n** : Machine learning techniques (black-box machine) using distributed and cloud computing.
 - ② **Big p** : High dimensional statistics ($p > n$) assuming some special structures.
 - ③ **Unstructured data**: text mining, pattern recognition, natural language processing (NLP), etc.

Big Data in Statistics

- What is big data from a statistical perspective?
 - n : number of observations; p : number of features (dimensionality).
- ① **Big n** : Machine learning techniques (black-box machine) using distributed and cloud computing.
 - ② **Big p** : High dimensional statistics ($p > n$) assuming some special structures.
 - ③ **Unstructured data**: text mining, pattern recognition, natural language processing (NLP), etc.

Introduction to Statistical Learning

Q1. What is Statistical learning?

Q2. Regression vs classification

Q3. Supervised learning vs unsupervised learning

What is Statistical Learning?

- Y : dependent variable, response variable, output.
- $\mathbf{X} = (X_1, \dots, X_p)^T$, regressors, covariates, features, independent variables, inputs.
- We can model the relationship as

$$Y = f(\mathbf{X}) + \varepsilon,$$

where f is an unknown function and ε captures measurement error (randomness) with mean zero.

- We use the **training data** $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ to estimate f .

The function $f(\mathbf{x})$

- Is $f(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$ optimal predictor of Y with regard to mean-squared prediction error?

The function $f(\mathbf{x})$

- Is $f(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$ optimal predictor of Y with regard to mean-squared prediction error?
- Yes, $f(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$ is the function that minimize $E[(Y - g(\mathbf{X}))^2 | \mathbf{X} = \mathbf{x}]$ over all functions g at all points $\mathbf{X} = \mathbf{x}$.
- $\varepsilon = Y - f(\mathbf{x})$ is the irreducible error, i.e. even if we knew $f(\mathbf{x})$, we would still make errors in prediction.

What is Statistical Learning?

- **Supervised learning:**
 - ① **Regression:** Y is continuous/numerical.
 - ② **Classification:** Y is categorical.
- We will deal with both problems.
 - ① Some methods work well on both types, e.g. Neural Networks.
 - ② Other methods work best on Regression, e.g. Linear Regression, or on Classification, e.g. k-Nearest Neighbors.

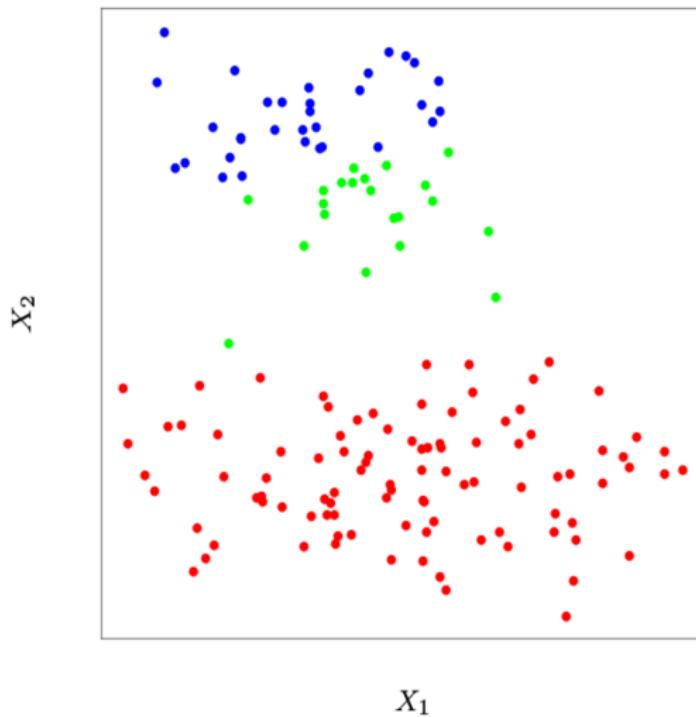
What is Statistical Learning?

- **Supervised learning:**
 - ① Regression: Y is continuous/numerical.
 - ② Classification: Y is categorical.
- We will deal with both problems.
 - ① Some methods work well on both types, e.g. Neural Networks.
 - ② Other methods work best on Regression, e.g. Linear Regression, or on Classification, e.g. k-Nearest Neighbors.
- Statistical learning, and this course, are all about how to estimate f .
- The term statistical learning refers to using data to “learn” f .
- Two reasons for estimating f
 - ① Prediction (predictive accuracy).
 - ② Inference (estimation accuracy + uncertainty).

Unsupervised Learning

- No outcome variable, just a set of features measured on a set of samples.
- **Clustering:** Find groups of samples that behave similarly;
- **Principal component analysis (PCA):** Find linear combinations of features with the most variation.
- e.g. Market segmentation: we try to divide potential customers into groups based on their characteristics.
- Difficult to know how well you are doing.
- We will consider unsupervised learning at the end of this course.

A Simple Clustering Example



Statistical Learning vs Machine Learning

- ① Machine learning is a subfield of Artificial Intelligence.
 - ML emphasis more on **large scale problems** and **prediction accuracy**.

Statistical Learning vs Machine Learning

- ① Machine learning is a subfield of Artificial Intelligence.
 - ML emphasis more on **large scale problems** and **prediction accuracy**.
 - ② Statistical learning is a subfield of Statistical Sciences.
 - SL emphasis more on building **statistical models** and their **estimation accuracy, uncertain** and **interpretability**.

Statistical Learning vs Machine Learning

- ① Machine learning is a subfield of Artificial Intelligence.
 - ML emphasis more on **large scale problems** and **prediction accuracy**.
 - ② Statistical learning is a subfield of Statistical Sciences.
 - SL emphasis more on building **statistical models** and their **estimation accuracy, uncertain** and **interpretability**.
-
- ① Statistical Learning vs Deep Learning?
 - ② Machine Learning and Artificial Intelligence in Finance?

Assessing Model Accuracy

Q4. How to assess the accuracy of our method?

Q5. How to choose the best one among all candidate methods?

Q6. What is the bias-variance tradeoff?

Q7. What is k-nearest neighbors for classification/regression?

Loss and Risk

- We use a **loss function**, L to quantify the prediction accuracy. We consider two types of loss functions

- ① ℓ_2 loss function for regression:

$$L(Y, f(X)) = (Y - f(X))^2.$$

- ② 0 – 1 loss function for classification:

$$L(Y, f(X)) = I(Y \neq f(X))$$

- For a given loss function, L , the **risk** of a learning function f is defined by the expected loss

$$R(f) = E_{X,Y}(L(Y, f(X))).$$

- We aim to find $f(x)$ that minimize $R(f)$ pointwise, the solution is, e.g.

- ① ℓ_2 loss: $f(x) = E(Y|X = x)$.

- ② ℓ_1 loss: $f(x) = \text{median}(Y|X = x)$.

Measuring the Quality of Fit to Data

- Suppose we observe i.i.d. samples $(x_i, y_i), i = 1, \dots, n$. The **empirical risk** is

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

We use training data to find \hat{f} , which $R_n(f)$.

- Regression:** Mean squared error (MSE) is given by

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2.$$

- Classification:** Misclassification error rate (MER) is given by

$$MER = \frac{1}{n} \sum_{i=1}^n I(y_i \neq \hat{f}(x_i)).$$

- In either case, our method has generally been designed to make MSE or the MER small on the training data we are looking at, e.g. linear regression, we choose **LSE!**

Testing Errors

- What we really care about is how well the method works on new data.
We call this new data **testing data**.
- We aim to choose the method that gives the lowest **test** MSE or MRE for regression and classification problems, respectively.

Testing Errors

- What we really care about is how well the method works on new data.
We call this new data **testing data**.
- We aim to choose the method that gives the lowest **test** MSE or MRE for regression and classification problems, respectively.
- Which one is more flexible?
 - ① Parametric model vs non-parametric model.
 - ② Linear model vs non-linear model.
 - ③ Linear model with 10 features vs linear model with 100 features.

Flexibility vs Interpretability

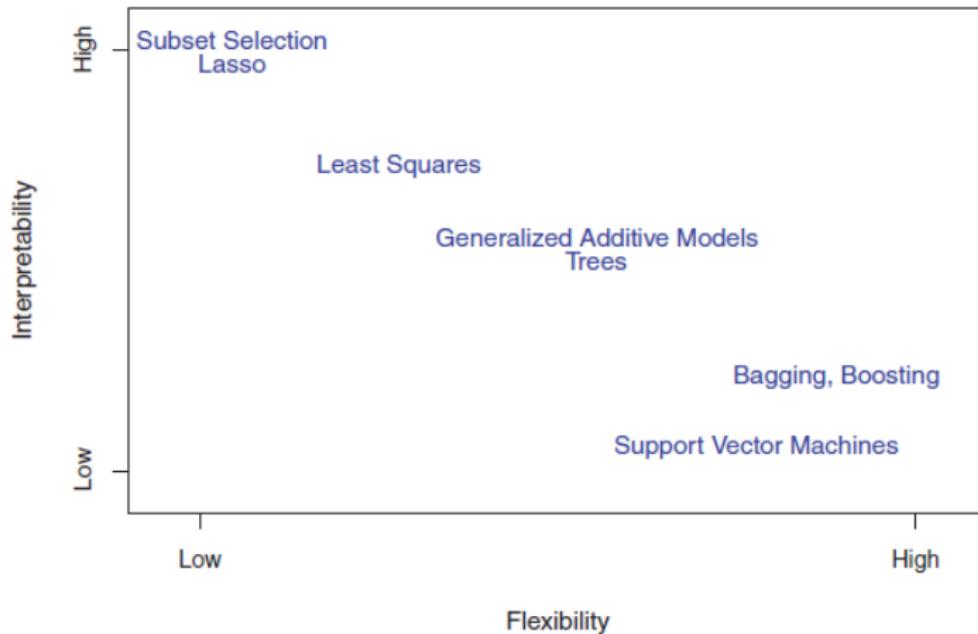
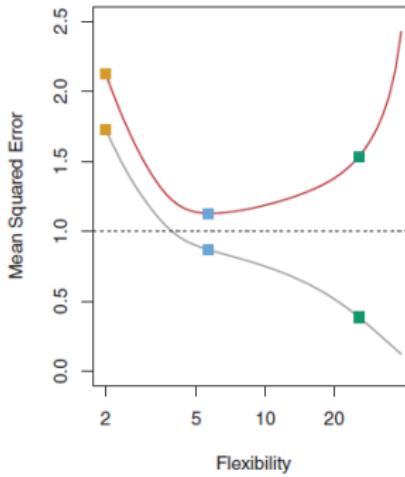
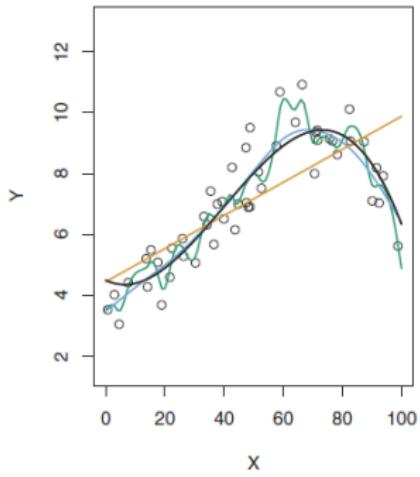
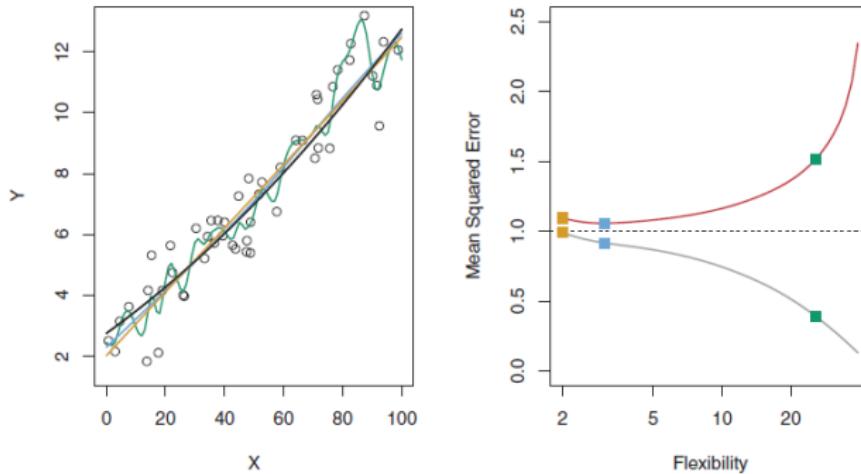


Figure: A representation of the tradeoff using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.

Training vs Testing Errors



Training vs Testing Errors



- In general, the more flexible a method is the lower its training error rate will be, i.e. it will “fit” or explain the training data very well.
- However, the test error rate may in fact be higher for a more flexible method than for a simple approach like linear regression.

Bias-Variance Tradeoff

- If the true model is $Y = f(X) + \varepsilon$, where $f(x) = E(Y|X=x)$.
- Suppose we have fitted a model $\hat{f}(x)$ based on training data and (x_0, y_0) be a test observation from the population. Then the expected test MSE is

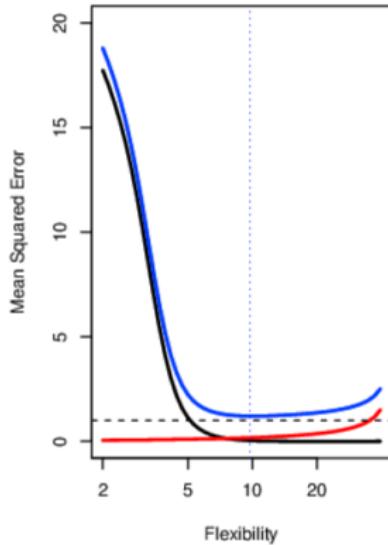
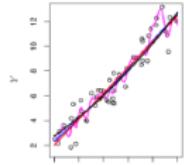
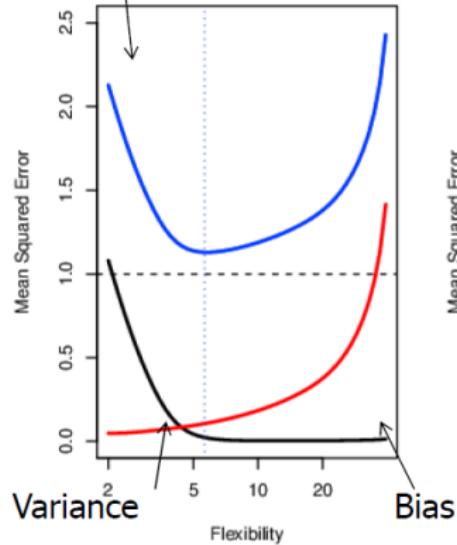
$$E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon),$$

where $\text{Bias}(\hat{f}(x_0)) = E[\hat{f}(x_0)] - f(x_0)$.

- As the **flexibility** of \hat{f} increases, its variance increases and its bias decreases. This corresponds to the **bias-variance tradeoff**.
- To minimize $E(y_0 - \hat{f}(x_0))^2$, we need to select a statistical learning method that simultaneously achieve relatively *low variance* and *low bias*.

Test Error, Bias and Variance

Test MSE



Classification Setting

- **Bayes classifier:** $f_{\text{Bayes}} : x \rightarrow \{1, \dots, J\}$ is the one with the minimum risk

$$\begin{aligned} E_Y(L(Y, f(X))|X = x) &= \sum_{j=1}^J L(j, f(x))P(Y = j|X = x) \\ &= P(Y \neq f(X)|X = x), \end{aligned}$$

where 0-1 loss function $L(Y, f(X)) = I(Y \neq f(X))$.

- The risk $1 - P(Y = f(X)|X = x)$ is minimized by choosing the class with the highest posterior probability

$$f_{\text{Bayes}} = \arg \max_{j=1, \dots, J} P(Y = j|X = x)$$

- The minimum risk attained by the Bayes classifier is called the **Bayes risk**.

The Bayes Error Rate

- The **Bayes error rate** refers to the lowest possible error rate that could be achieved if somehow we knew exactly what the true probability distribution of the data looked like,

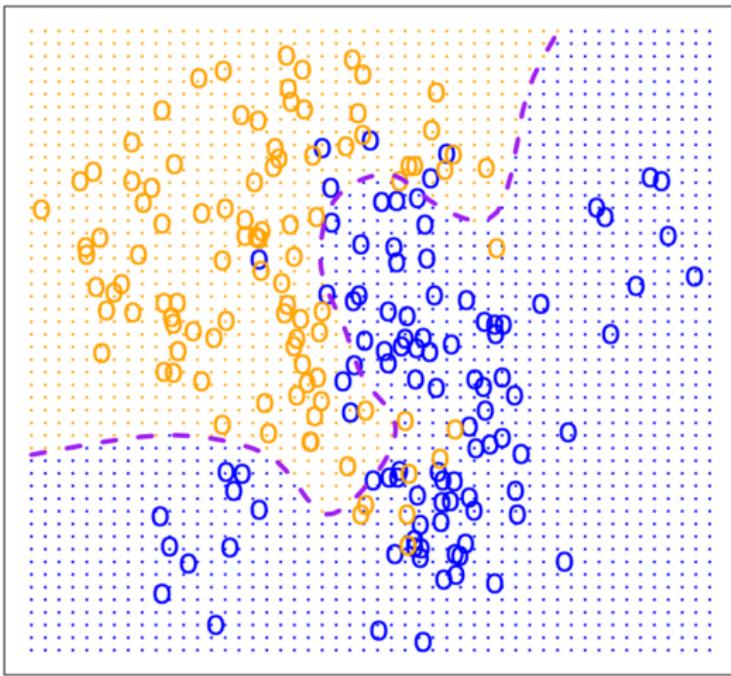
$$e_{\text{Bayes}}(x) = 1 - \max_{j=1,\dots,J} P(Y = j | X = x).$$

- The **overall Bayes error rate** is

$$E_X(e_{\text{Bayes}}(X)).$$

- On test data, no classifier can get lower error rates than the Bayes error rate.
- Of course in real life problems the Bayes error rate can not be calculated exactly.

Bayes Optimal Classifier



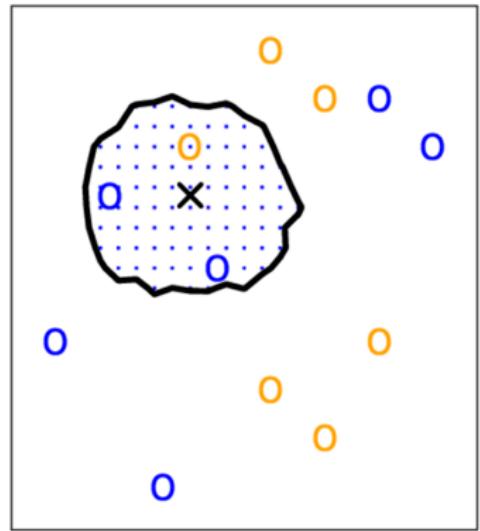
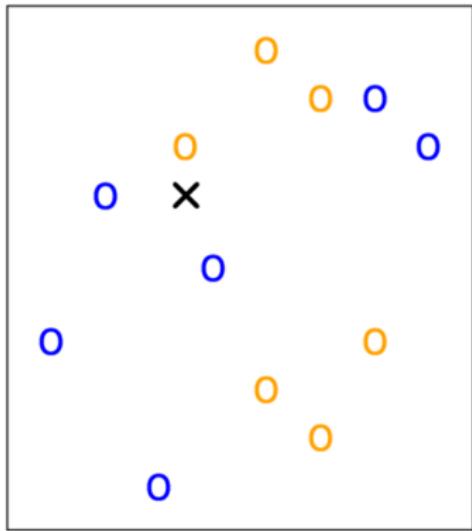
k-Nearest Neighbors (kNN)

- $P(Y|X = x)$ is unknown for real data and many approaches attempt to estimate the conditional probability.
- Given a positive integer k and a test observation x_0 , **kNN classifier** first identifies k points in the training data that are closest to x_0 , represented by \mathcal{N}_0 whose response values equal j :

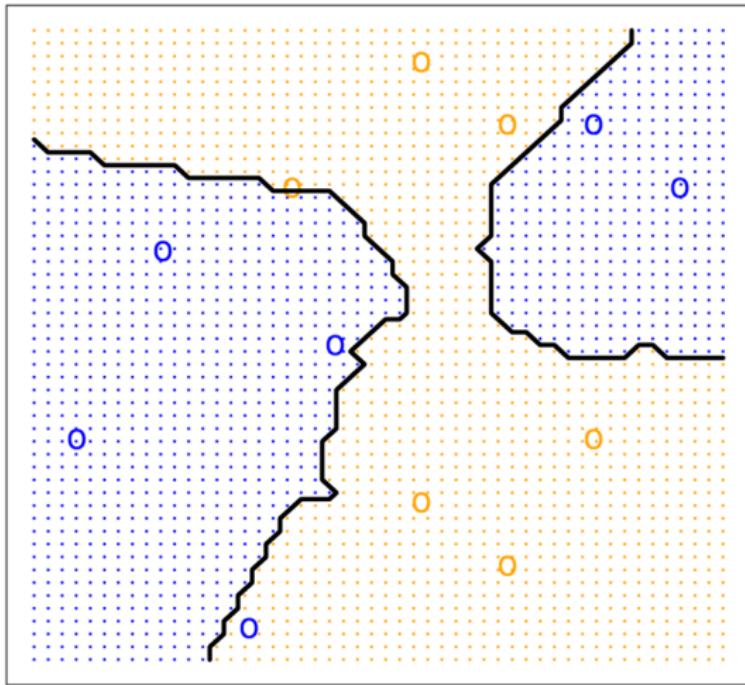
$$P(Y = j|X = x_0) = \frac{1}{k} \sum_{i \in \mathcal{N}_0} I(y_i = j).$$

- If the majority of the Y 's are orange we predict orange otherwise guess blue.
- The smaller the k , the ?? the method will be.

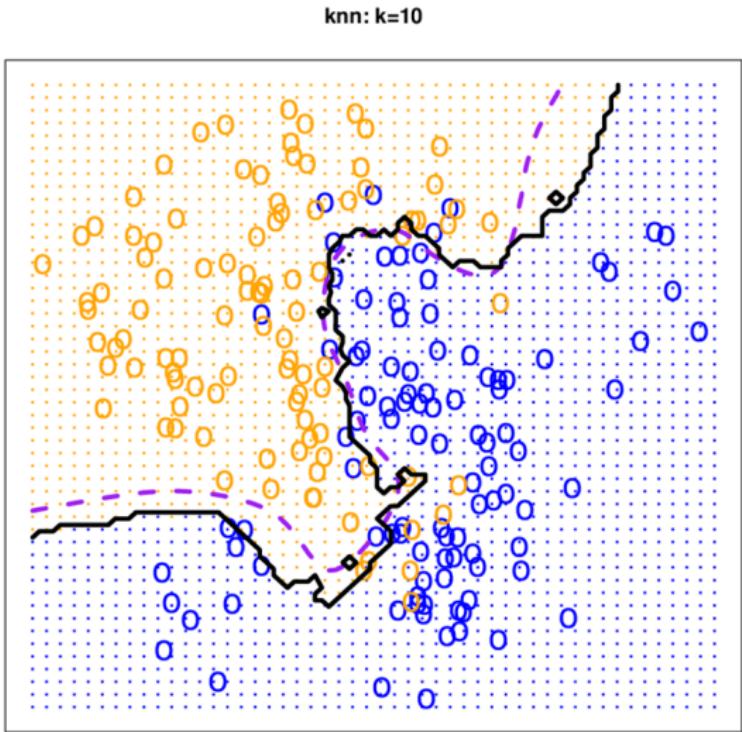
kNN Example with $k = 3$



kNN Decision Boundary

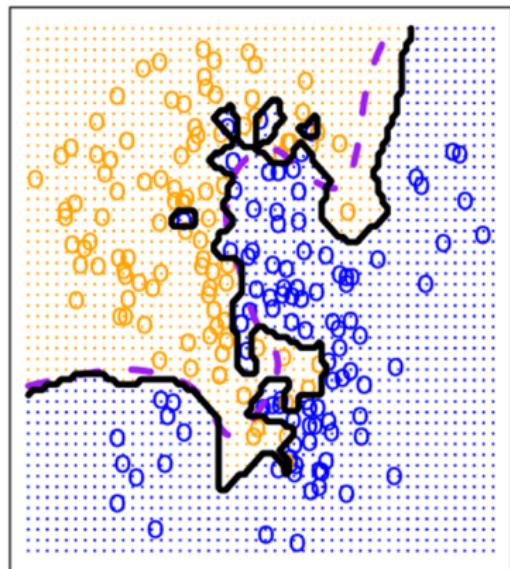


Simulated Data: $k = 10$

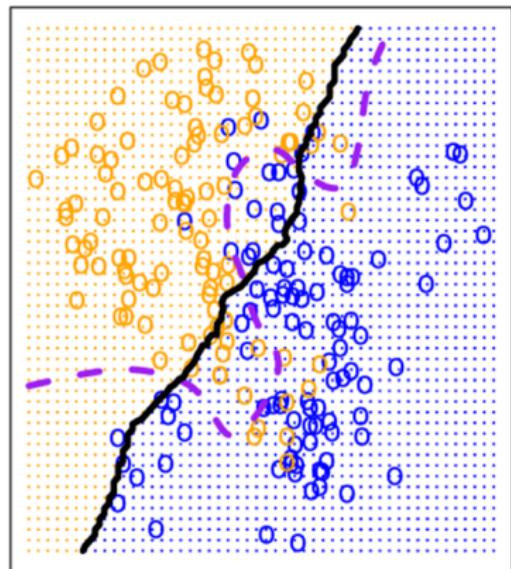


$k = 1$ and $k = 100$

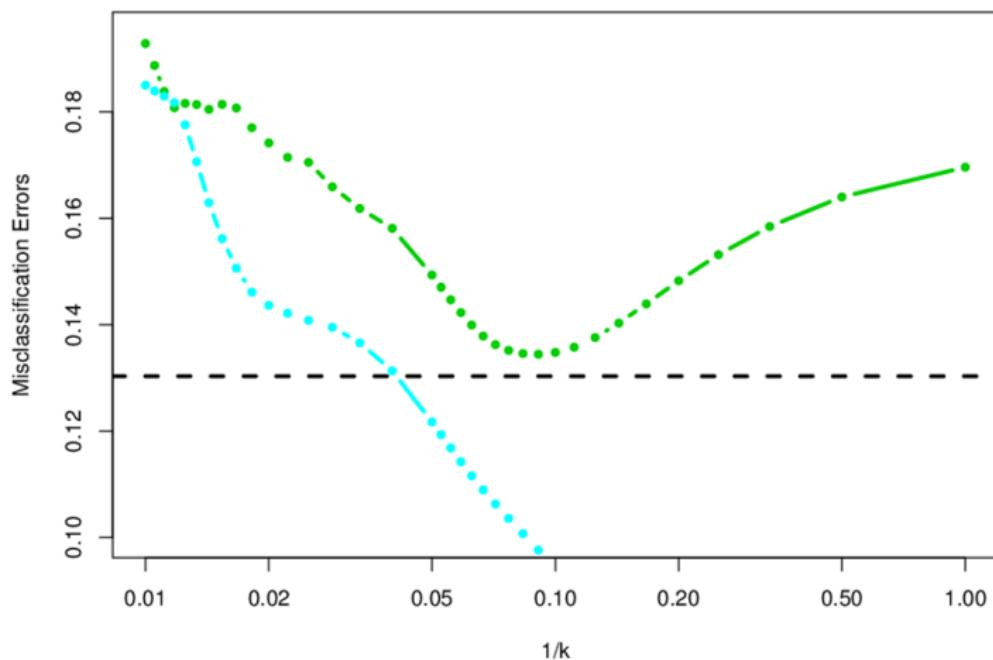
knn: k=1



knn: k=100



Training vs Testing Error Rates on Simulated Data



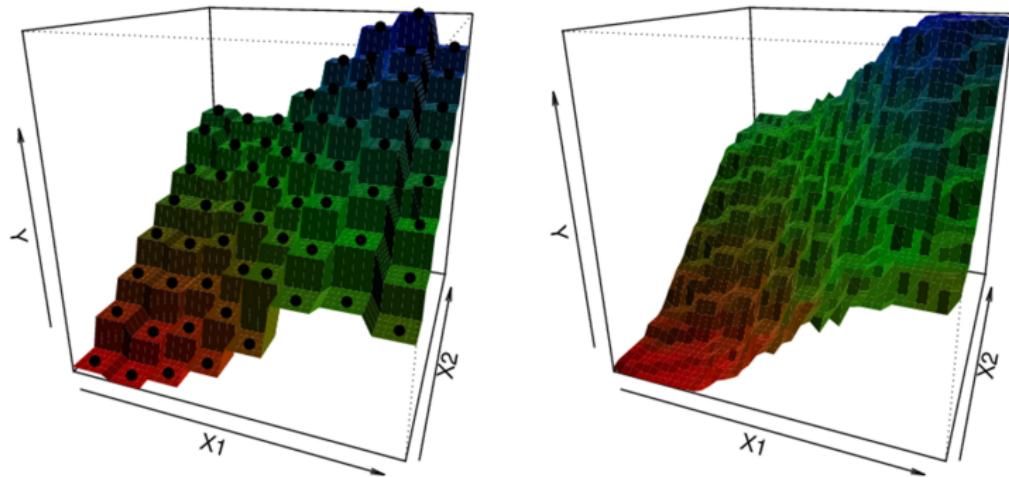
kNN Regression

- kNN regression is similar to kNN Classifier.
- Given a value for k and a prediction point x_0 , kNN regression first identifies the k training observations that are closest to x_0 , represented by \mathcal{N}_0 and predict

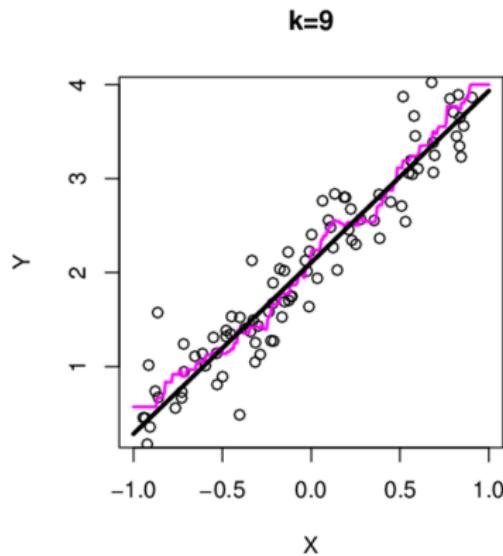
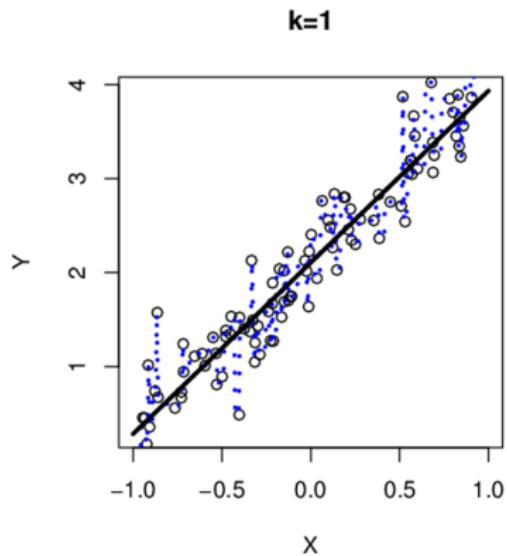
$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in \mathcal{N}_0} y_i.$$

- If k is small, kNN is much more flexible than linear regression. Is that better?

kNN Fits for $k = 1$ and $k = 9$

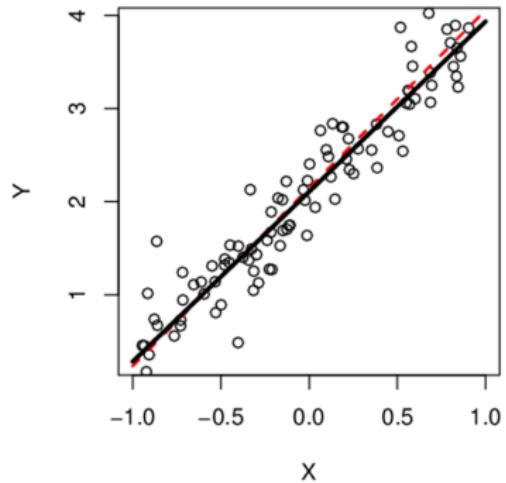


kNN Fits in One Dimension

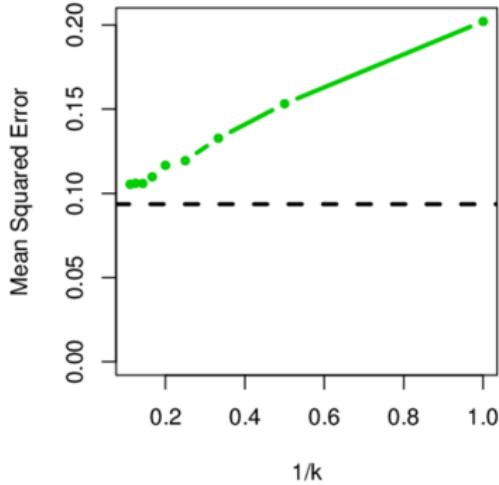


Linear Regression Fit

Linear Regression

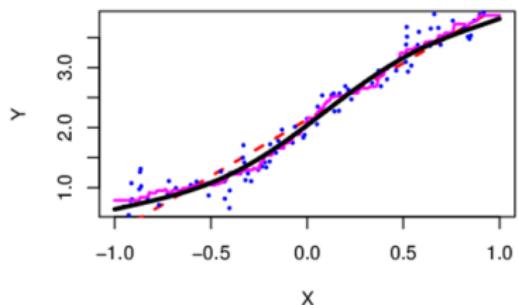


MSE under Linearity

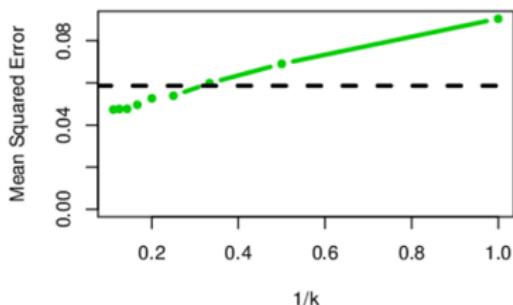


kNN vs Linear Regression

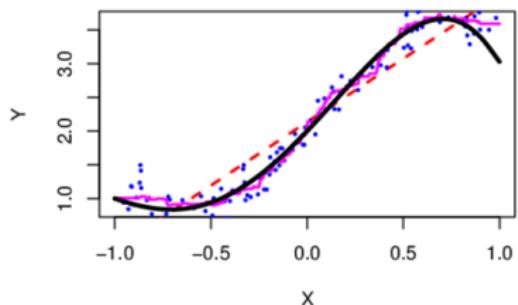
Mild Non-Linearity



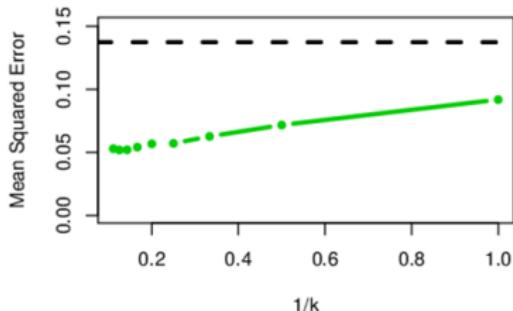
MSE under Mild Non-Linearity



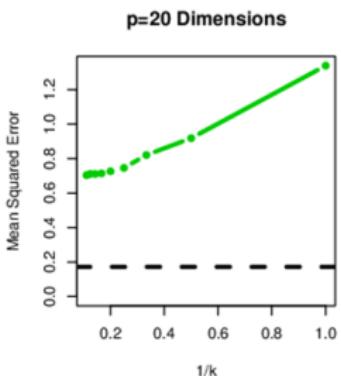
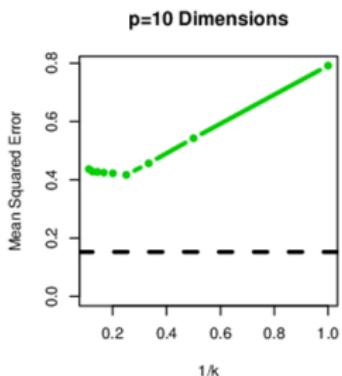
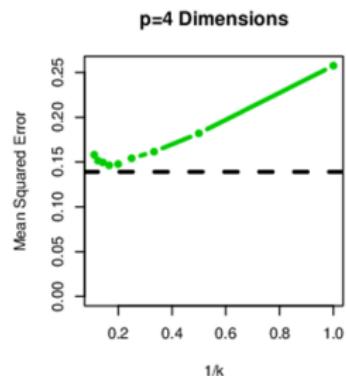
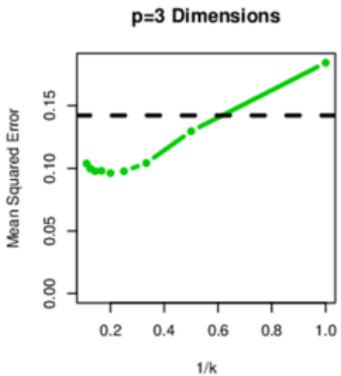
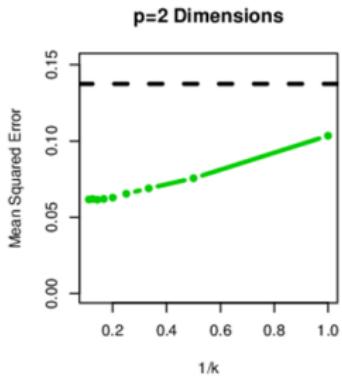
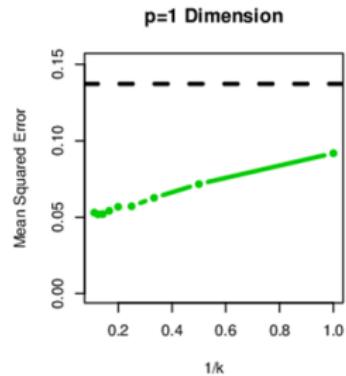
Strong Non-Linearity



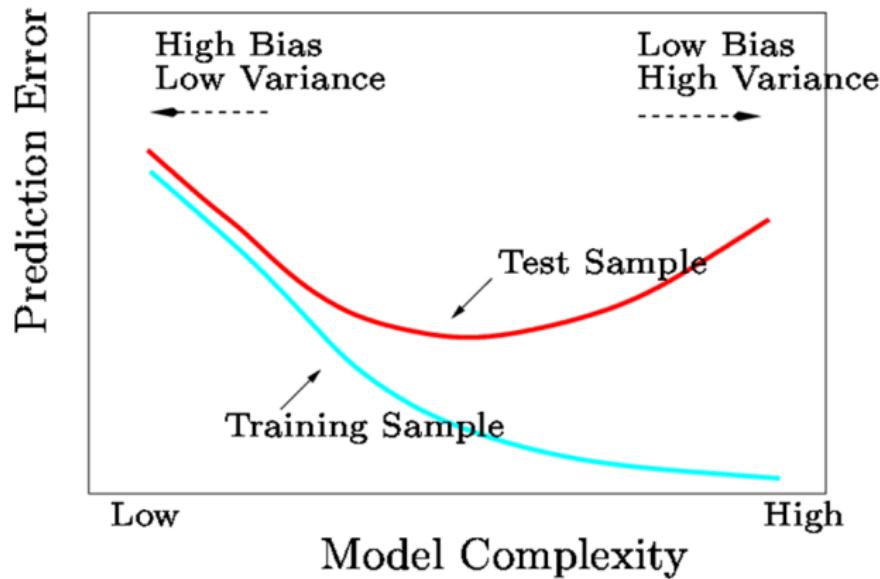
MSE under Strong Non-Linearity



Not So Good in High Dimensional Situations



A Fundamental Picture



We must keep this picture in mind when choosing a learning method. More flexible/complicated one is not always better!

ST 443: Machine Learning and Data Mining

Dr. Xinghao Qiao

Columbia House, Room 5.15

X.Qiao@lse.ac.uk

Office Hours: Tuesday 4:30–5:30pm
Department of Statistics



TA: Cheng Chen, C.Chen44@lse.ac.uk. OH: COL 5.03, Thu 4-5pm,
TA: Yirui Liu, Y.Liu110@lse.ac.uk. OH: COL 5.03, Thu 9-10am

Lecture 2

Review of Lecture 1

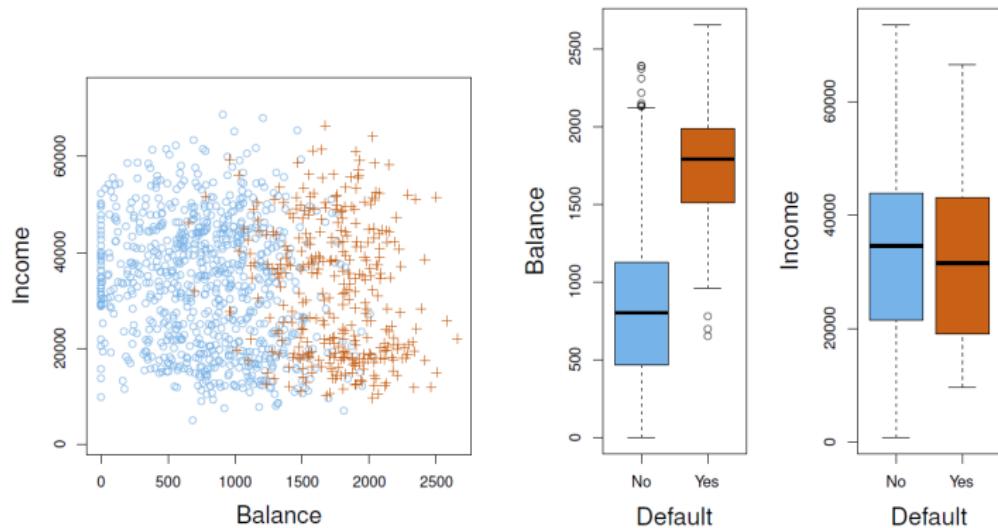
- Definition of statistical learning
- Regression vs classification
- Supervised vs unsupervised learning
- Loss and risk
- Bias-variance tradeoff
- Bayes classifier
- KNN for classification and regression

Review of Linear Regression

- Please read Chapter 3 of ISL or lecture notes for ST 300.
- Topics include
 - Simple and multiple linear regression
 - Hypothesis testing
 - Model checking
 - Qualitative predictors
 - Interactions and nonlinearity
 - ...

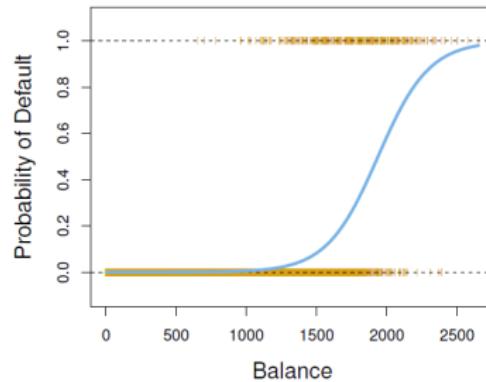
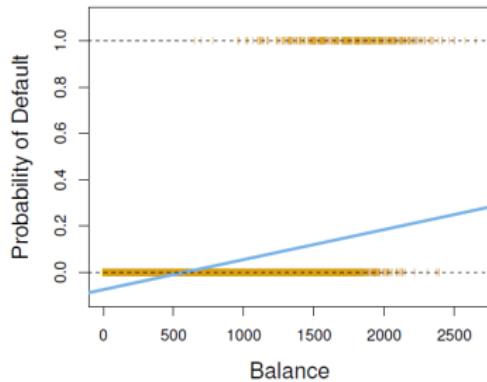
- Q1. Two class problem and logistics regression.**
- Q2. Linear and quadratic discriminant analysis.**
- Q3. A comparison of classification methods.**

Credit Card Default Data



- We let $Y = 0$ if No and 1 if Yes.
- Can we use a linear regression of Y on X and classify based on whether $\hat{Y} > 0.5$?

Linear vs Logistic Regression



- Orange marks denote $Y = 0$ or 1 .
- Although $E(Y|X) = P(Y = 1|X)$ given binary outcomes, linear regression does not estimate $P(Y = 1|X)$ well.
- **Logistic regression** uses

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \in [0, 1].$$

Logistic Regression

- Rewrite the model

$$\log \left(\frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X.$$

- This monotone transformation is called the **log odds** or **logit** transformation of $p(X)$.
- MLE for β_0, β_1 :

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

- If β_1 is positive then increasing X will be associated with increasing $p(X)$.
- We use **glm** function in R:

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

Making Predictions

- Suppose an individual has an average balance of 1000, what is the probability of default?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 1000}}{1 + e^{-10.6513 + 0.0055 \times 1000}} = 0.006$$

- What about with a balance of 2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 2000}}{1 + e^{-10.6513 + 0.0055 \times 2000}} = 0.586$$

Multi-Class Logistic Regression for $p > 1$

- The generalization to multi-class logistic regression takes the form

$$P(Y = k|X) = \frac{e^{\beta_{0k} + \beta_{1k}X_1 + \dots + \beta_{pk}X_p}}{\sum_{k=1}^K e^{\beta_{0k} + \beta_{1k}X_1 + \dots + \beta_{pk}X_p}}.$$

- Multinomial regression.
- Connection to 2-class logistic regression, any problem?

The Bayes Classifier

- **Bayes classifier:** $f_{\text{Bayes}} : x \rightarrow \{1, \dots, K\}$ is the one with the minimum risk,

$$E_Y(L(Y, f(X))|X = x) = \sum_{k=1}^K L(k, f(x))P(Y = k|X = x).$$

- Consider the 0-1 loss function for $L(Y, f(X))$, we obtain

$$f_{\text{Bayes}} = \operatorname{argmax}_{k=1, \dots, K} P(Y = k|X = x).$$

Discriminant Analysis

- The density g of X can be written as a mixture of K classes:

$$f(x) = \sum_{k=1}^K \pi_k f_k(x),$$

where $P(Y = k) = \pi_k$ represents the **prior probability** for class k and $f_k(x)$ denotes the **conditional density** of X given $Y = k$.

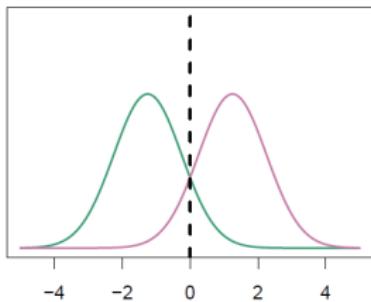
- Using Bayes Theorem, we use maximal **posterior probability**

$$\operatorname{argmax}_{k=1,\dots,K} P(Y = k|X = x) = \operatorname{argmax}_{k=1,\dots,K} \frac{\pi_k f_k(x)}{\sum_{k=1}^K \pi_k f_k(x)} = \operatorname{argmax}_{k=1,\dots,K} \pi_k f_k(x).$$

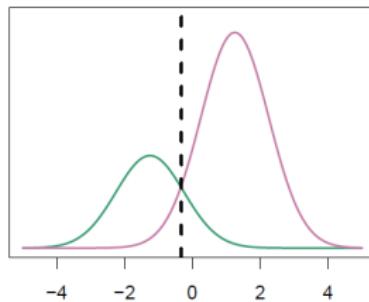
- $\pi_k f_k(x)$'s are called **discriminant function**.

Classify to the Highest Density

$$\pi_1 = .5, \quad \pi_2 = .5$$



$$\pi_1 = .3, \quad \pi_2 = .7$$



- Consider two class classification problem, assume $X \sim N(\mu_Y, \sigma^2)$, where $\mu_1 = -2, \mu_2 = 2$ with equal prior probability $\pi_1 = \pi_2 = 0.5$. What is the Bayes classifier using 0-1 loss function?
- When the priors are different, we compare $\pi_k f_k(x)$. In the right figure, the decision boundary has shifted to the left.
- In practice, π_k 's and $f_k(x)$'s are unknown. We need to estimate them from the training data.

Linear Discriminant Analysis (LDA) when $p = 1$

- For Gaussian distributed X , we have

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2},$$

where $X|Y=k \sim N(\mu_k, \sigma_k^2)$ and assume that $\sigma_k = \sigma, k = 1, \dots, K$.

- Using Bayes Theorem, we have

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

Linear Discriminant Analysis (LDA) when $p = 1$

- For Gaussian distributed X , we have

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2},$$

where $X|Y=k \sim N(\mu_k, \sigma_k^2)$ and assume that $\sigma_k = \sigma, k = 1, \dots, K$.

- Using Bayes Theorem, we have

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

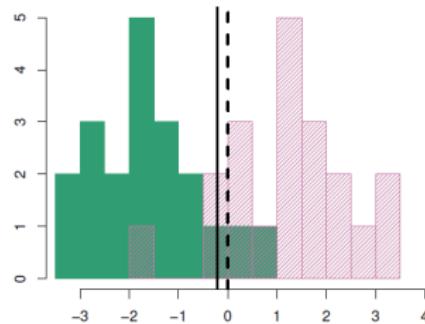
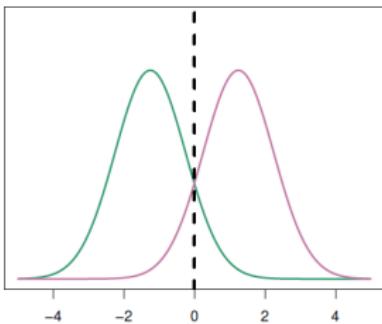
- We assign x to the class with the largest **discriminant score** over $\{1, \dots, K\}$ (**taking logs and remove terms that do not depend on k**)

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

which is a **linear** function of x .

- For $K = 2$ and $\pi_1 = \pi_2 = 0.5$ what is the **Bayes decision boundary**?

Estimating LDA when $p = 1$



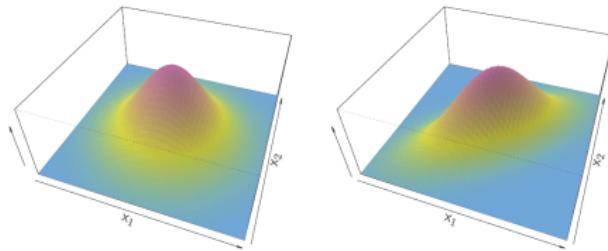
$$\hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n - K} \sum_{k=1}^K \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2 = \sum_{k=1}^K \frac{n_k - 1}{n - K} \hat{\sigma}_k^2,$$

where $\hat{\sigma}_k^2$ is the estimated variance in the k -th class.

LDA when $p > 1$



- Assume $\mathbf{X}|Y = k \sim MVN(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, $k = 1, \dots, K$

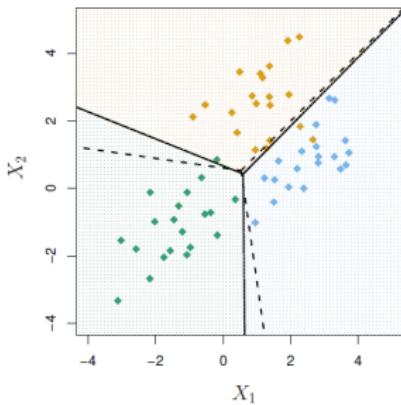
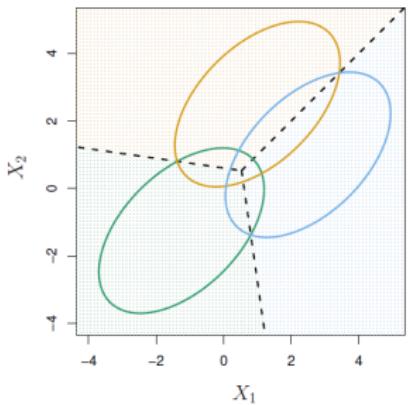
$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)}.$$

- Discriminant score (taking logs of $\pi_k f_k(\mathbf{x})$ and remove terms that do not depend on k):

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

is a linear function of $\mathbf{x} = (x_1, \dots, x_p)^T$.

Parameters estimation



- $p = 2$ and $K = 3$, Bayes error rate: 0.0746, LDA error rate=0.0770.
- The dashed lines are **Bayes decision boundaries**, which yield lowest misclassification error rates among all possible classifiers.
- Estimated parameters:

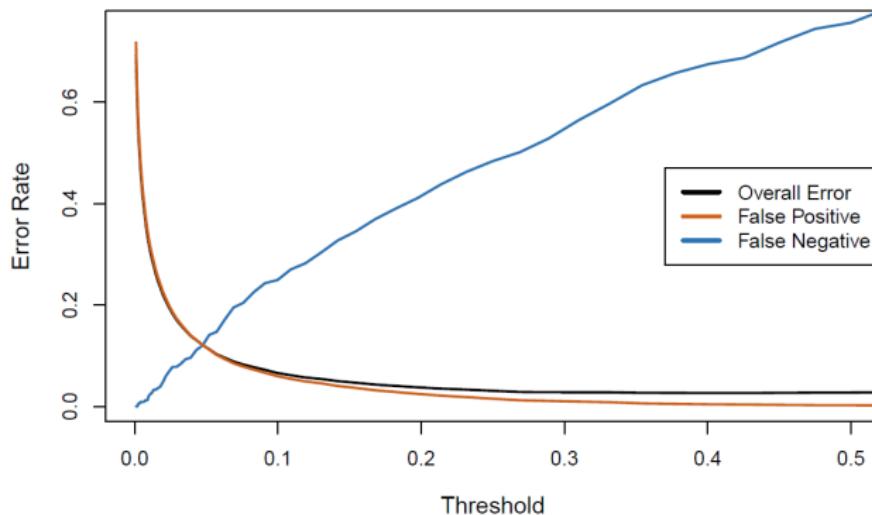
$$\hat{\pi}_k = \frac{n_k}{n}, \hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} \mathbf{x}_i, \hat{\Sigma} = \frac{1}{n-K} \sum_{k=1}^K \sum_{i:y_i=k} (\mathbf{x}_i - \hat{\mu}_k)(\mathbf{x}_i - \hat{\mu}_k)^T$$

Confusion Matrix under LDA on Credit Data

		True defaults		Total
		No	Yes	
LDA predictions	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

- If we classify based on prior information: always to class **No**, we make $333/10000=3.33\%$ errors.
- **False positive rate:** under true **No**'s, we make $23/9667=0.2\%$ errors.
- **False negative rate:** under true **Yes**'s, we make $252/333=75.7\%$ errors.
- Perhaps we should not use 50% as threshold for predicting default?

Default Errors vs Probability Threshold



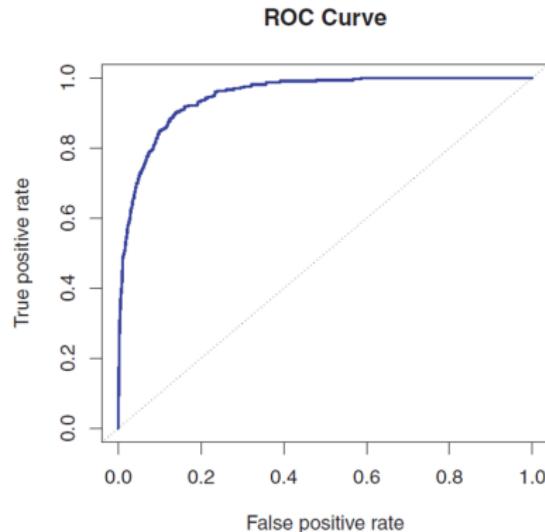
- To reduce FNR, we consider reduce the threshold.

Possible Results

		Predicted class		
		- or Null	+ or Non-null	Total
True class	- or Null	True Neg (TN)	False Pos (FP)	N
	+ or Non-null	False Neg (FN)	True Pos (TP)	P
	Total	N*	P*	

- False positive rate (FPR): FP/N (type I error, 1-specificity).
- True positive rate (TPR): TP/P (1-type II error, power, sensitivity).

ROC Curves



- ROC curve for LDA classifier on training data.
- AUROC: area under the curve, the larger AUROC the better overall performance.
- ROC curve provides a graphical illustration of two types of errors for all possible thresholds.

Discriminant Analysis

$$P(Y = k | \mathbf{X} = \mathbf{x}) = \frac{\pi_k f_k(\mathbf{x})}{\sum_{k=1}^K \pi_k f_k(\mathbf{x})}$$

- ① $f_k(x)$'s are multivariate Gaussian with the same covariance matrix $\Sigma_k = \Sigma$, we get [Linear Discriminant Analysis \(LDA\)](#).
- ② With different Σ_k in each class, we get [Quadratic Discriminant Analysis \(QDA\)](#).

Quadratic Discriminant Analysis (QDA)

- Assume $\mathbf{X}|Y = k \sim MVN(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), k = 1, \dots, K$

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)}.$$

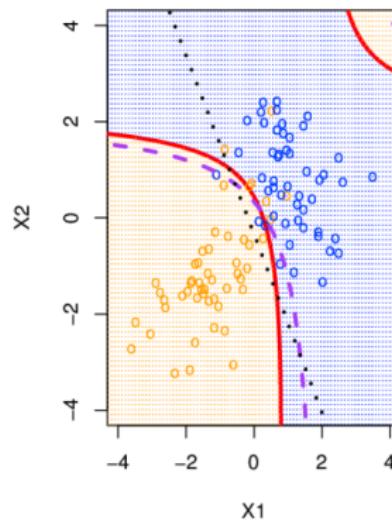
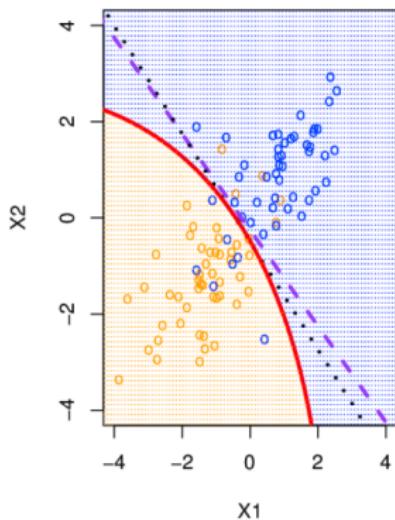
- Discriminant score:

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

is a quadratic function of $\mathbf{x} = (x_1, \dots, x_p)^T$.

QDA vs LDA

- Which approach is better LDA or QDA?
- QDA will work better when Σ_k 's are very different between classes and we have enough observations to accurately estimate Σ_k 's.
- LDA will work better when Σ_k 's are similar among classes or we do not have enough data to accurately estimate Σ_k 's.



Bayes = Purple Dashed, LDA = Black dotted, QDA = Red solid

Logistic Regression vs LDA

For two-class problem, we can show that LDA reduces to

$$\log \left(\frac{p_1(x)}{1 - p_1(x)} \right) = \log \left(\frac{p_1(x)}{p_2(x)} \right) = c_0 + c_1 x_1 + \cdots + c_p x_p.$$

- **Similarity**

- ① Both approaches produce linear boundaries.
 - ② In practice, the results are very similar.

- **Difference**

- ① Logistic regression uses the conditional likelihood $P(Y|X)$ (**discriminative learning**).
 - ② LDA uses full likelihood $P(X, Y)$ (**generative learning**).
 - ③ In practice, the results are very similar.
 - ④ Logistic regression can still produce quadratic boundaries. How?

KNN vs LDA and Logistic Regression

- KNN takes a completely different approach.
- KNN is completely non-parametric: no assumptions are made about the shape of the decision boundary.
- **Advantage of KNN:** We can expect KNN to dominate both LDA and logistic regression when the decision boundary is highly non-linear.
- **Disadvantage of KNN:** KNN does not tell us which predictors are important (no table of coefficients).

QDA vs LDA, Logistic Regression and KNN

- QDA is a compromise between non-parametric KNN method and LDA and logistic regression.
- Logistic regression can also fit quadratic boundaries by explicitly including quadratic terms.
- If the true decision boundary is
 - linear: LDA and logistic regression outperforms.
 - moderately non-linear: QDA outperforms.
 - more complicated: KNN is superior.
- Other flexible classification approaches are to be introduced.

ST 443: Machine Learning and Data Mining

Dr. Xinghao Qiao

Columbia House, Room 5.15

x.qiao@lse.ac.uk

Department of Statistics



Office Hours: Tuesday 4:30–5:30pm

Lecture 3

Resampling Methods

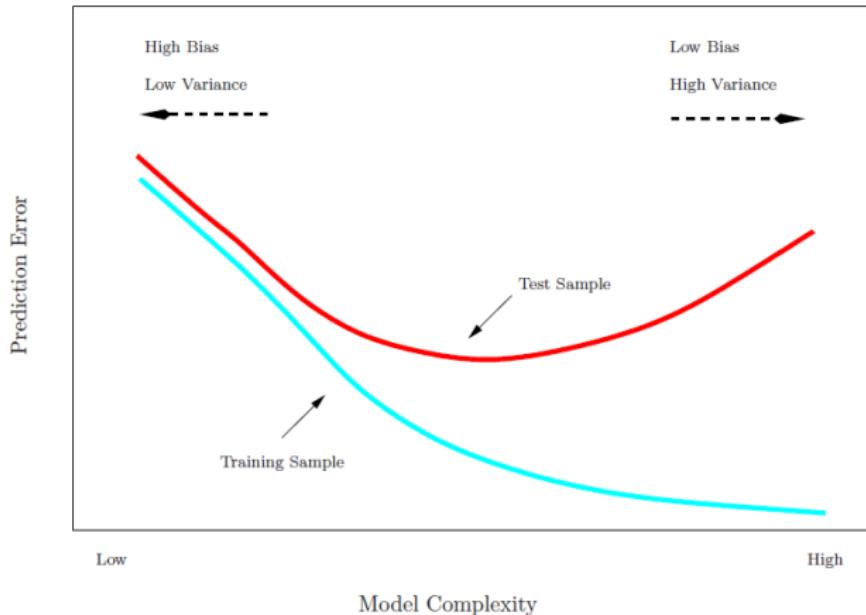
Q1. Cross Validation.

Q2. The Bootstrap.

What are Resampling Approaches?

- Tools that involves repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain more information about the fitted model.
 - Model assessment: estimate test error rates.
 - Model selection: select the appropriate level of model flexibility.
- They are computationally expensive! But computers can help us!
- Two resampling approaches.
 - Cross Validation.
 - Bootstrap.

What are Resampling Approaches?



Ways to Estimate the Test Errors

- Best solution, a large designed test set. Not available!
- Mathematical adjustments to the training error rate to estimate the test error rate, e.g. C_p , statistics, AIC and BIC (to be discussed).
- Resampling approaches: estimate the test error by holding out a subset of the training observations and applying the statistical learning approaches to those held-out observations.

Typical Approach: The Validation Set Approach

- Suppose that we would like to find a set of variables that give the lowest **test error rate**.
- Given a large data set, we can randomly split the data into training set and validation set or hold-out set.
- The model is fit on the training set, and the fitted model is used to predict the response for the observations in the validation set. The resulting validation set error rate, typically assessed using MSE, provides an estimate of the **test error rate**.

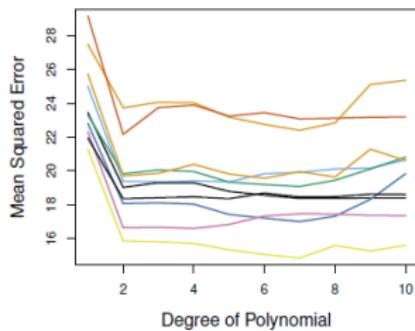
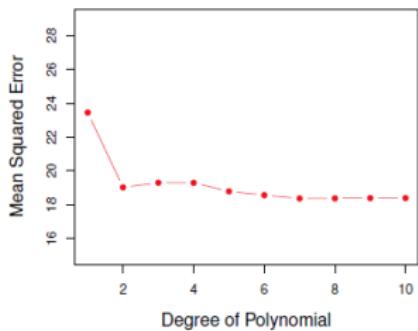


Example: Auto Data

- Suppose that we want to predict mpg from horsepower.
- Two models:
 - ① $\text{mpg} \sim \text{horsepower}$
 - ② $\text{mpg} \sim \text{horsepower} + \text{horsepower}^2$
- Which model gives a better fit?
 - ① Randomly split Auto data into training (196 obs.) and validation data (196 obs.)
 - ② Fit both models using the training data set.
 - ③ Then evaluate both models using the validation data set.
 - ④ The model with the lowest validation (testing) MSE is the winner!

Results: Auto Data

- Left: validation error rate for a single split.
- Right: validation method repeated 10 times, each time the split is done randomly.
- There is **large variability** among the MSE's. **Not good!** We need more stable method.

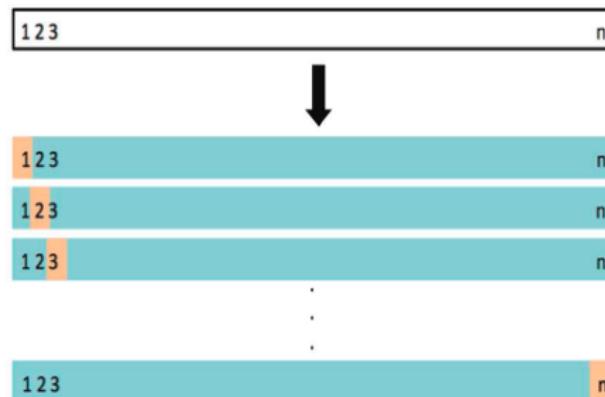


The Validation Set Approach

- Advantages:
 - ① Simple.
 - ② Easy to implement.
- Disadvantages:
 - ① The validation MSE can be **highly variable**.
 - ② Only a subset of observations are used to fit the model (training data). Statistical methods tend to perform worse when trained on fewer observations, which suggests the validation set error rate may tend to **overestimate** the test error rate for the model fit on the entire data set.

Leave-One-Out Cross Validation (LOOCV)

- This method is similar to the Validation Set Approach, but it tries to address latter disadvantages.
- For each suggested model, do:
 - ① Split the data set of size n into: training data set (blue) size: $n-1$ and validation data set (beige) size: 1.
 - ② Fit the model using the training data.
 - ③ Validate model using the validation data, compute the corresponding MSE.
 - ④ Repeat the process n times.
 - ⑤ The Model MSE: $CV_{(n)}(\hat{f}) = \frac{1}{n} \sum_{i=1}^n \text{MSE}_i = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}^{(-i)}(x_i))^2$



LOOCV vs the Validation Set Approach

- LOOCV has less bias
 - We repeatedly fit the statistical learning method using training data that contains $n - 1$ observations.
 - LOOCV tends not to overestimate the test error rate.
- LOOCV produces a less variable MSE
 - The validation approach produces different MSE when applied repeatedly due to randomness in the splitting process, while performing LOOCV multiple times will always yield the same results. (**Why?**)
- LOOCV is computational intensive (disadvantage).
 - We fit each model n times.

Shortcut in “Linear” Models

- Let $\mathbf{y} = (y_1, \dots, y_n)^T$ and similarly for the predictions $\hat{\mathbf{y}}$, then a linear fitting method is one for which we can write

$$\hat{\mathbf{y}} = \mathbf{S}\mathbf{y},$$

where \mathbf{S} is a $n \times n$ matrix depending on \mathbf{x}_i but not on y_i .

- For many linear fitting methods,

$$CV_{(n)}(\hat{f}) = \frac{1}{n} \sum_{i=1}^n \left(y_i - \hat{f}^{(-i)}(\mathbf{x}_i) \right)^2 = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{f}(\mathbf{x}_i)}{1 - S_{ii}} \right)^2,$$

where S_{ii} is the i -th diagonal element of \mathbf{S} .

- The Generalized Cross Validation (GCV) provides an approximation to LOOCV,

$$GCV(\hat{f}) = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{f}(\mathbf{x}_i)}{1 - \text{trace}(\mathbf{S})/n} \right)^2.$$

- $\text{trace}(\mathbf{S})$ is the effective number of parameters and sometimes be computed easily than S_{ii} 's. **Example?**

K-fold Cross Validation (CV)

- LOOCV is computational intensive, so we can run K -fold CV instead.
- With K -fold CV, we divide the data set into K parts (e.g. $K=5$ or 10 , rule of thumb), C_1, \dots, C_K , where C_k denotes the indices of the observations in part k . There are n_k observations in part k .
- Compute

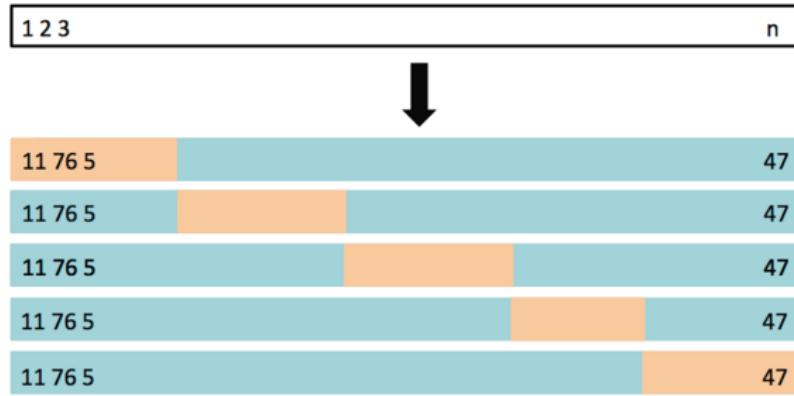
$$CV_{(K)} = \sum_{k=1}^K \frac{n_k}{n} \text{MSE}_k,$$

where $\text{MSE}_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$ and \hat{y}_k is the fit for observation i , obtained from the data with part k removed.

- Mathematically, let $\kappa : \{1, \dots, n\} \rightarrow \{1, \dots, K\}$. Denote by \hat{f}^{-k} the fitted function, computed with the k th part of the data removed. Then

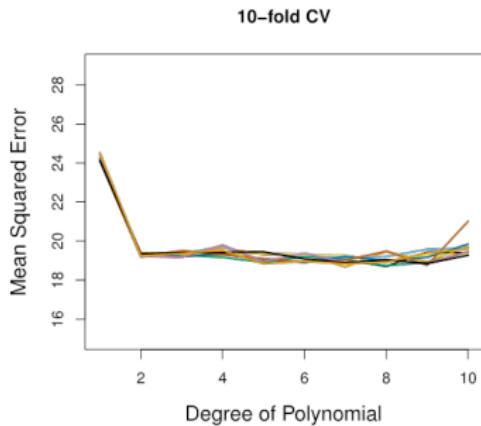
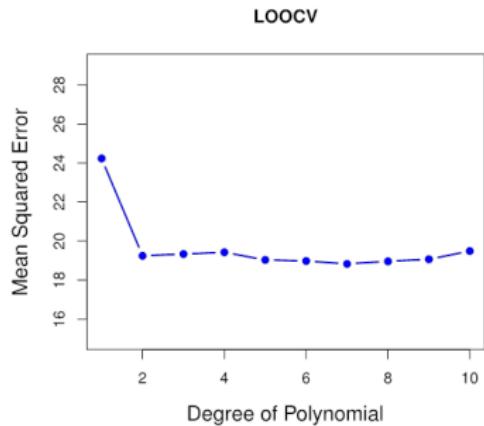
$$CV_{(K)}(\hat{f}) = \frac{1}{n} \sum_{i=1}^n L \left(y_i, \hat{f}^{-\kappa(i)}(\mathbf{x}_i) \right).$$

K -fold CV



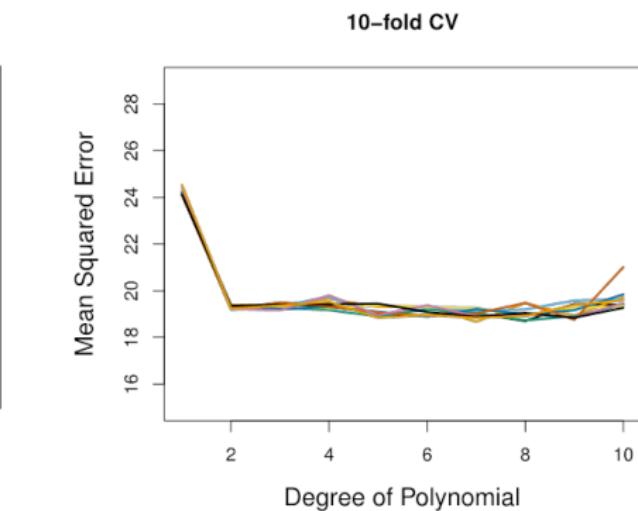
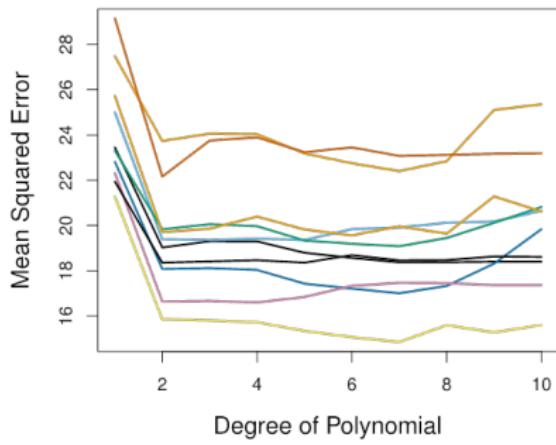
Auto Data: LOOCV vs K -fold CV

- LOOCV is a special case of K -fold CV, where $K = n$.
- They are both stable, but LOOCV is more computational intensive.



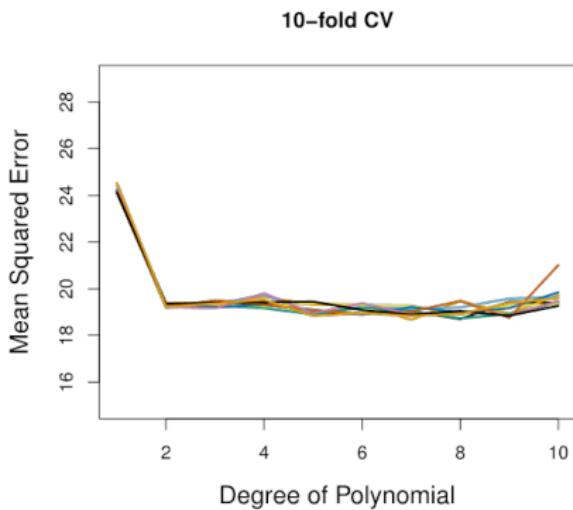
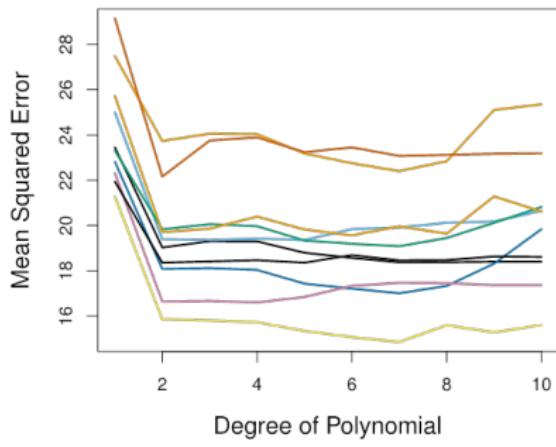
Auto Data: Validation Set Approach vs K-fold CV

- K -fold CV is more stable.



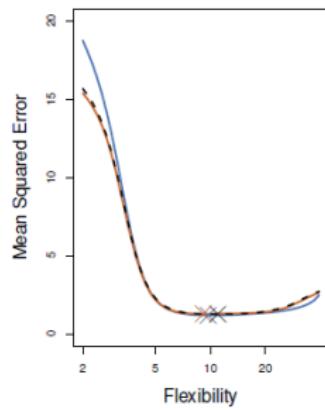
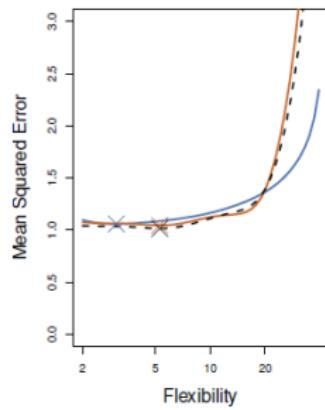
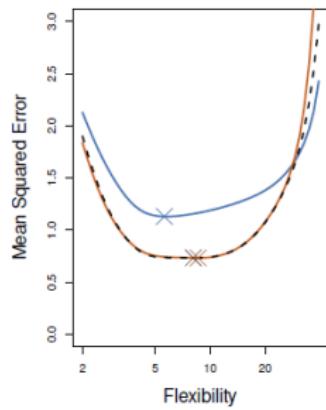
Auto Data: Validation Set Approach vs K-fold CV

- K -fold CV is more stable.



K-fold CV on Three Simulated Data

- Blue: True test MSE.
- LOOCV MSE.
- Orange: 10-fold CV.



Variance and Bias Tradeoff with CV Approaches

- Putting aside that LOOCV is more computational intensive than K -fold CV. Which one is better LOOCV or K -fold CV?

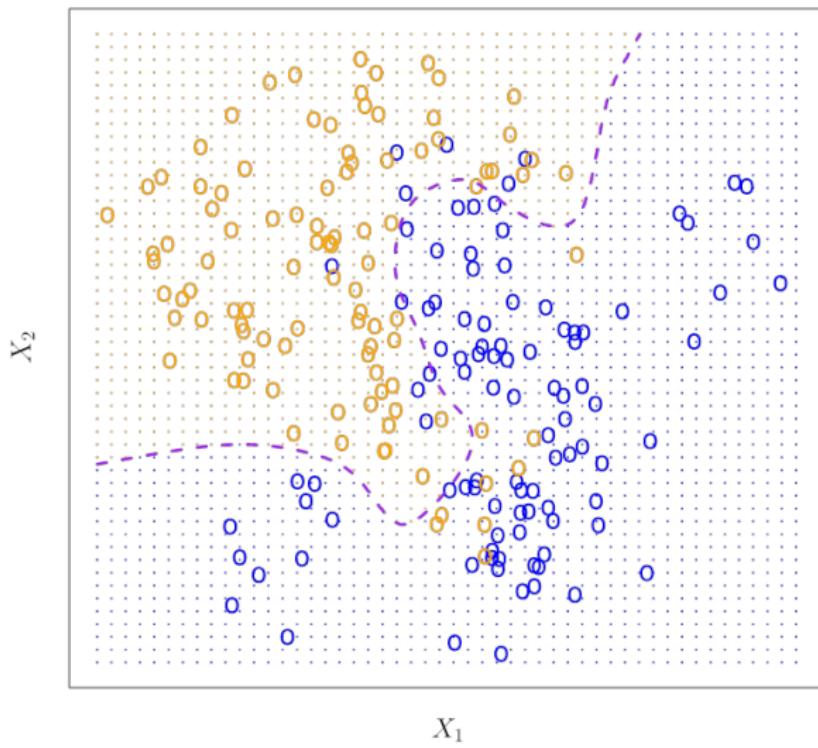
Variance and Bias Tradeoff with CV Approaches

- Putting aside that LOOCV is more computational intensive than K -fold CV. Which one is better LOOCV or K -fold CV?
 - Since each training is only $(K - 1)/K$ as big as the original training data, the estimates of prediction error will typically be biased upwards.
 - LOOCV has higher variance than K -fold CV.
 - There is a tradeoff between what to use.

Variance and Bias Tradeoff with CV Approaches

- Putting aside that LOOCV is more computational intensive than K -fold CV. Which one is better LOOCV or K -fold CV?
 - Since each training is only $(K - 1)/K$ as big as the original training data, the estimates of prediction error will typically be biased upwards.
 - LOOCV has higher variance than K -fold CV.
 - There is a tradeoff between what to use.
- Conclusion:
 - To compromise, we use K -fold CV with $K=5$ or 10 .
 - It has been empirically shown that they yield test error rate estimates that suffer neither from excessively high bias, nor from very high variance.

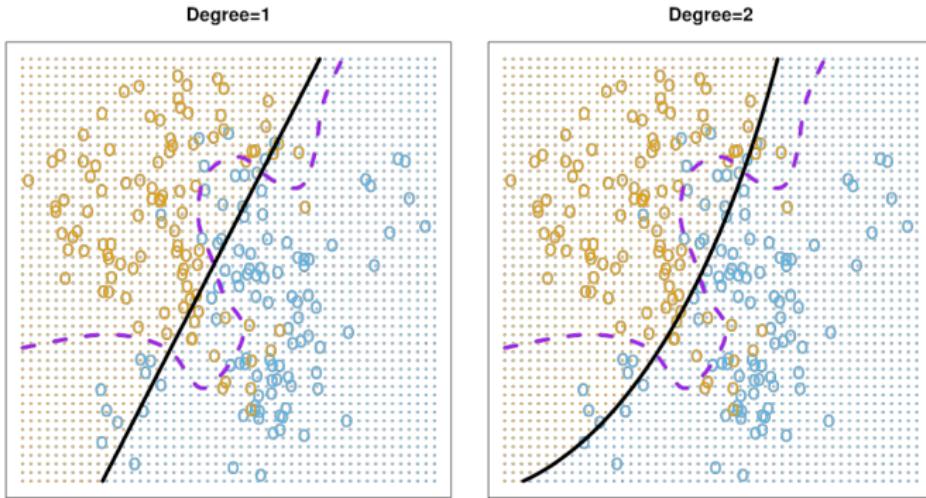
CV on Classification Problems



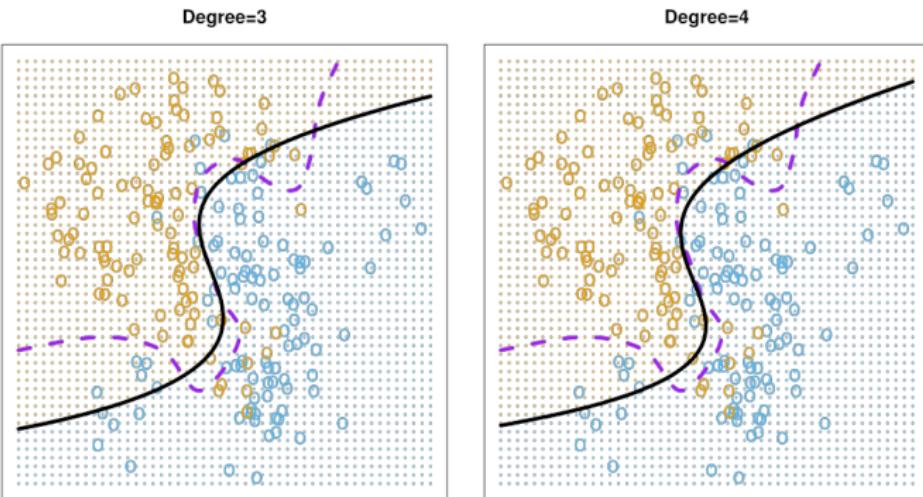
Bayes error rate: 0.133.

Bayes Error Rate: 0.133

Logistic regression is used.

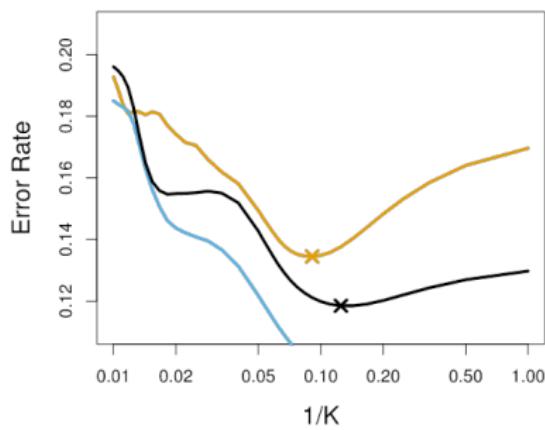
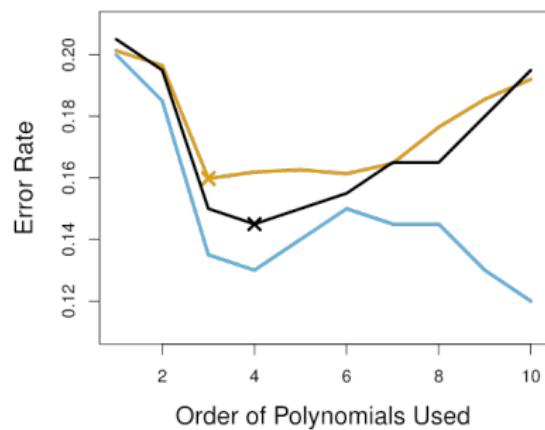


Bayes Error Rate: 0.133



CV to Choose the Order

Brown: test error, blue: training error, black: 10-fold CV error.



Another Resampling Approach: Bootstrap

- Bootstrap is widely applicable and extremely powerful statistical tool to quantify the uncertainty associated with the given estimator or statistical learning method.
- Linear model, generalized linear model, standard errors can be computed (mathematical representation or R output).
- Bootstrap can be easily applied to a wide range of approaches including for which a measure of variability is difficult to calculate or be obtained from software.

A Toy Example: Best Investment Allocation

- Two financial assets that yields returns X and Y .
- We invest a fraction α and $1 - \alpha$ in X and Y .
- Problem: choose α to minimize the risk.

$$\text{Var}(\alpha X + (1 - \alpha) Y).$$

- The value of α that minimize the risk is given by

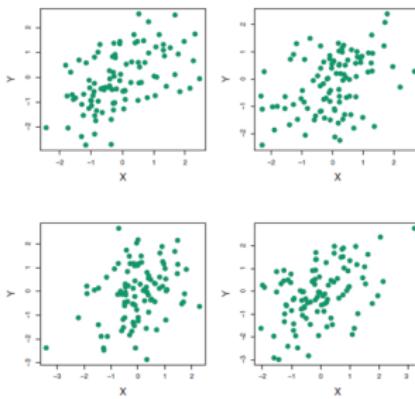
$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

, where $\sigma_X^2 = \text{Var}(X)$, $\sigma_Y^2 = \text{Var}(Y)$ and $\sigma_{XY} = \text{Cov}(X, Y)$.

- We estimate α using

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

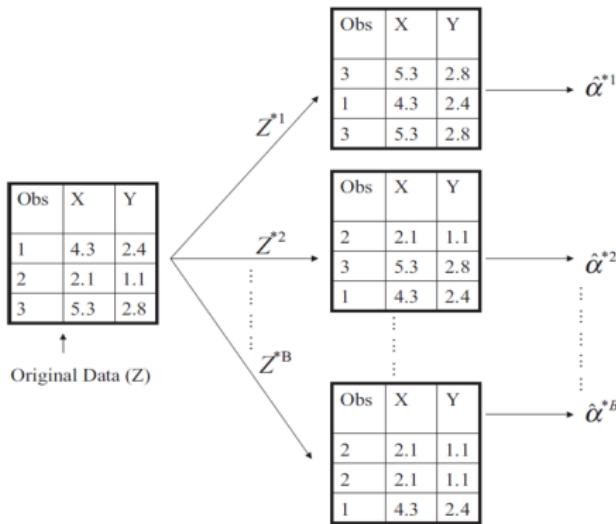
Sampling Approach with Known Population Distribution



- How to quantify the estimate for α ?
- We repeat the process of simulating 100 paired observations X and Y ($n = 100$) and estimate for 1000 times, $\hat{\alpha}_1, \hat{\alpha}_2, \dots, \hat{\alpha}_{1000}$.
- Set $\sigma_X^2 = 1, \sigma_Y^2 = 1.25$ and $\sigma_{XY} = 0.5$, so $\alpha = 0.6$.
- The mean $\bar{\alpha} = \frac{1}{1000} \sum_{b=1}^{1000} \hat{\alpha}_b = 0.5996$.
- The standard error $\sqrt{\frac{1}{1000-1} \sum_{b=1}^{1000} (\hat{\alpha}_b - \bar{\alpha})^2} = 0.083$

A Bootstrap Approach

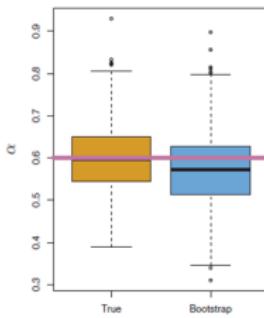
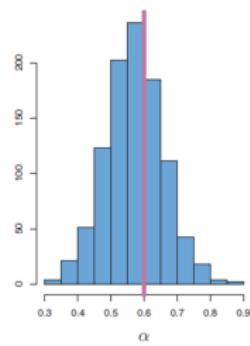
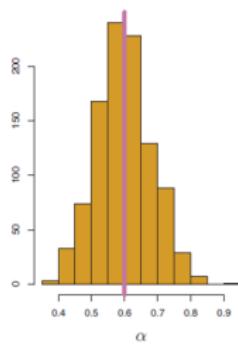
- Rather than repeatedly obtaining independent data sets from the population (unknown), we obtain distinct data sets by repeatedly sampling observations from the original data set.
- One simple example with 3 observations and sampling is performed with replacement,



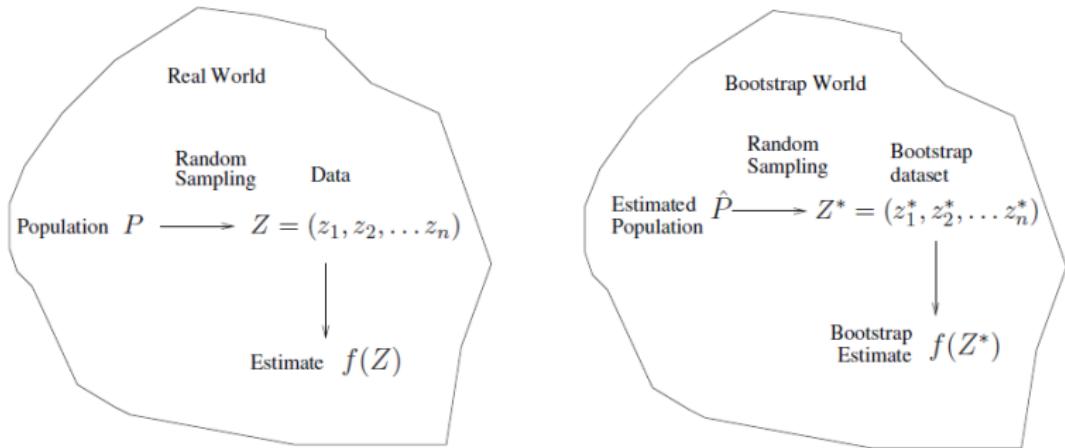
Bootstrap Procedure

- The sampling (with replacement) procedure is repeated B times to produce B different bootstrap data sets Z^{*1}, \dots, Z^{*B} , and B corresponding α estimates $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \dots, \hat{\alpha}^{*B}$.
- We can compute the standard error of the bootstrap estimates by

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{b=1}^B \left(\hat{\alpha}^{*b} - \frac{1}{B} \sum_{b'=1}^B \hat{\alpha}^{*b'} \right)^2}$$



A General Picture for the Bootstrap



Other Uses of the Bootstrap

- Mainly to obtain the standard errors of an estimate.
- Provides approximate confidence intervals for a population parameter.
E.g. revisiting the histogram, the 5% and 95% quantiles of 1000 values are (0.43, 0.72).
- This provides an approximate 90% confidence interval for α ,
bootstrap percentage confidence interval.

Other Uses of the Bootstrap

- Mainly to obtain the standard errors of an estimate.
- Provides approximate confidence intervals for a population parameter.
E.g. revisiting the histogram, the 5% and 95% quantiles of 1000 values are (0.43, 0.72).
- This provides an approximate 90% confidence interval for α ,
bootstrap percentage confidence interval.

Can the Bootstrap Estimate Prediction Error? (TBD)

- One approach is to fit the model in question on a set of bootstrap samples and then keep track of how well it predicts the original training set. Let $\hat{f}^{*b}(x_i)$ be the predicted value at x_i from the model fitted to the b th bootstrap sample, our estimate is

$$\widehat{\text{Err}}_{\text{boot}} = \frac{1}{Bn} \sum_{b=1}^B \sum_{i=1}^n L(y_i, \hat{f}^{*b}(x_i))$$

- Each bootstrap has significant overlap with the original data. About **2/3 of the original data points in each bootstrap sample.** Why?

Can the Bootstrap Estimate Prediction Error? (TBD)

- One approach is to fit the model in question on a set of bootstrap samples and then keep track of how well it predicts the original training set. Let $\hat{f}^{*b}(x_i)$ be the predicted value at x_i from the model fitted to the b th bootstrap sample, our estimate is

$$\widehat{\text{Err}}_{\text{boot}} = \frac{1}{Bn} \sum_{b=1}^B \sum_{i=1}^n L(y_i, \hat{f}^{*b}(x_i))$$

- Each bootstrap has significant overlap with the original data. About **2/3 of the original data points in each bootstrap sample.** Why?

$$P(\text{obs } i \in \text{bootstrap sample } b) = 1 - \left(1 - \frac{1}{n}\right)^n \approx 1 - e^{-1} = 0.632.$$

- This will cause the bootstrap to seriously underestimate the true prediction error.

.632 Estimator (TBD)

- A better bootstrap estimate can be obtained. For each observation, we only keep track of predictions from bootstrap samples not containing that observation.
- The leave-one-out bootstrap estimate of prediction error is defined by

$$\widehat{\text{Err}}^{(1)} = \frac{1}{n} \sum_{i=1}^n \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} L\left(y_i, \hat{f}^{*b}(x_i)\right),$$

where C^{-i} is the set of indices of the bootstrap samples b that do not obtain observation i . Any problem?

.632 Estimator (TBD)

- A better bootstrap estimate can be obtained. For each observation, we only keep track of predictions from bootstrap samples not containing that observation.
- The **leave-one-out bootstrap estimate** of prediction error is defined by

$$\widehat{\text{Err}}^{(1)} = \frac{1}{n} \sum_{i=1}^n \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} L(y_i, \hat{f}^{*b}(x_i)),$$

where C^{-i} is the set of indices of the bootstrap samples b that do not obtain observation i . **Any problem?**

- The **.632 estimator** is designed to alleviate this bias. Define

$$\widehat{\text{Err}}^{(.632)} = .368\overline{\text{err}} + .632\widehat{\text{Err}}^{(1)},$$

where $\overline{\text{err}} = \frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i))$.

ST 443: Machine Learning and Data Mining

Dr. Xinghao Qiao

Columbia House, Room 5.15

x.qiao@lse.ac.uk

Department of Statistics



Office Hours: Tuesday 4:30–5:30pm

Lecture 4

Q1. Subset selection

Q2. Shrinkage methods

Q3. Dimension reduction methods

A General Picture

- Recall the linear model

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p + \varepsilon$$

- Two reasons one might prefer not to use OLS estimates
 - ① Prediction accuracy
 - ② Model interpretability

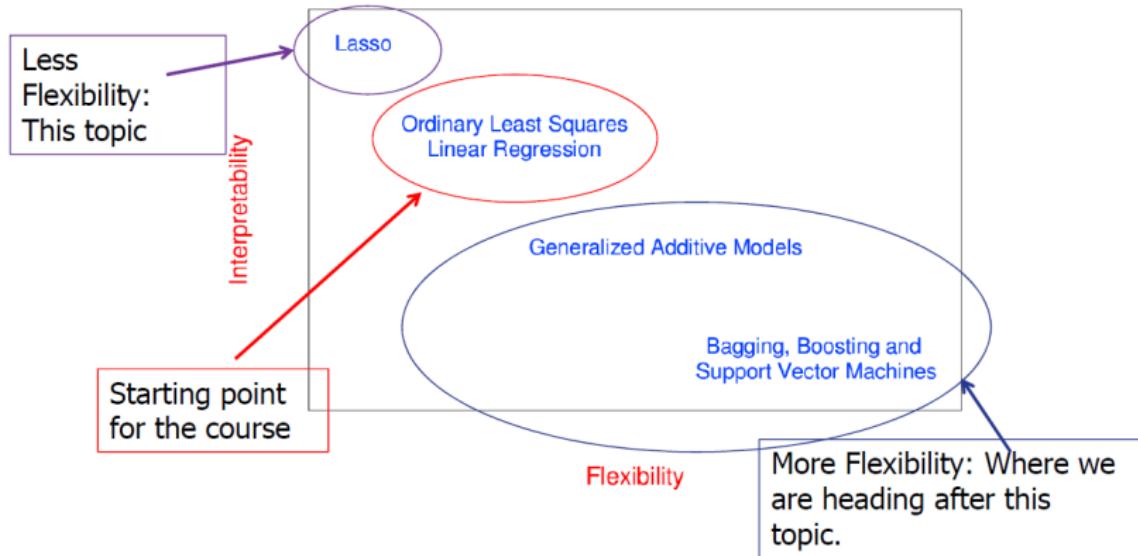
Prediction Accuracy

- The **LSE** have relatively low bias and low variability especially when the relationship between Y and X is linear and the number of observations n is way bigger than the number of predictors p
- But, when n is as large as p , then least squares fit can have high variance and may result in over-fitting and poor prediction on unseen observations.
- And, when $n < p$, the variability of the least squares fit increases dramatically, and the variance of these estimates is infinite.

Model Interpretability

- When we have a large number of variables X in the model there will generally be many that have little or no effect on Y .
- Leaving these variables in the models makes it harder to see the “big picture”, i.e. the effect of “important variables.”
- By removing irrelevant features, i.e. setting the corresponding coefficient estimates to zero, we obtain a model that is more easily interpreted.
- We will present some approaches for automatically performing variable selection!

Flexibility vs Interpretability



Three Types of Approaches

- **Subset Selection.** We identify a subset of p predictors that we believe to be related to the response. Then we fit a model using least squares on the selected set of variables.
- **Shrinkage.** We fit a model including all p predictors, and the estimated coefficients using least squares are shrunken towards zero. This shrinkage reduces variance and automatically perform variable selection.
- **Dimension Reduction.** We project p predictors into a M -dimensional subspace ($M < p$) through calculating M different linear combinations of the variables. We then regress the response on those M projections using least squares.

Best Subset Selection

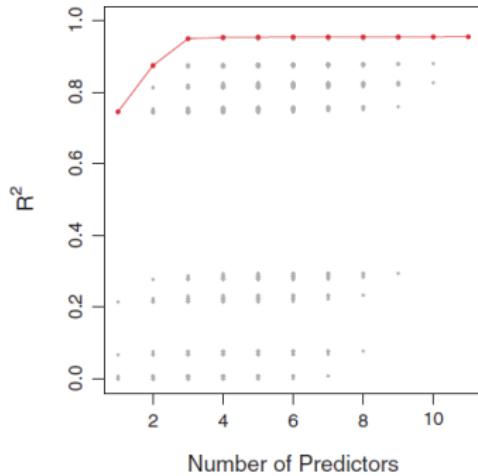
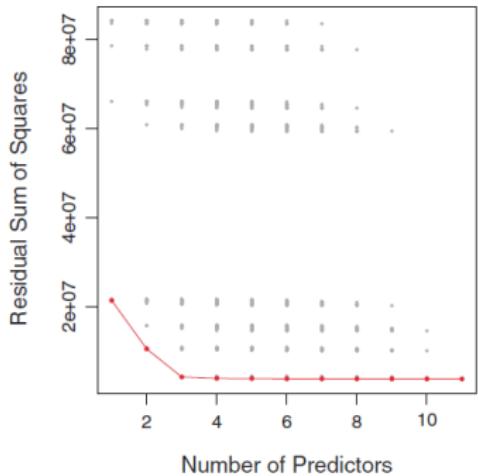
Forward/Backward Subset Selection

Choosing the Optimal Model

Best Subsect 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 *residual sum of squares* (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 .

Credit Data Example



Stepwise Selection

- **Computational** problem: best subset selection is not a feasible approach when p is very large.
- **Statistical** problem: the larger the search space, the higher the possibility of selecting models that look pretty good on the training data, but have poor performance on prediction for new inputs.
(Overfitting and high variance from bias-variance trade-off aspect.)
- **Stepwise methods**, which perform the search on a restricted set of variables, is needed to overcome the above two disadvantages.

Forward Stepwise Selection (FSS)

- ① Let \mathcal{M}_0 denote the *null* model, which contains no predictors.
- ② For $k = 1, 2, \dots, p - 1$:
 - a Fit all $p - k$ models that augment the predictors in \mathcal{M}_k with one additional predictor.
 - b Pick the *best* among these $p - k$ models, and call it \mathcal{M}_{k+1} . 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 .

Pros and Cons of FSS

- Computational advantageous to the best subset selection approach
- FSS approach searches through $1 + p(p + 1)/2$ models.
- It is not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors. **Why? Any example?**

Credit Data Example

# Variables	Best subset	Forward stepwise
One	rating	rating
Two	rating, income	rating, income
Three	rating, income, student	rating, income, student
Four	cards, income student, limit	rating, income, student, limit

The first four selected models for best subset selection and forward stepwise selection on the Credit data set. The first three models are identical but the fourth models differ.

Backward Stepwise Selection (BSS)

- 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 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 .

More about BSS

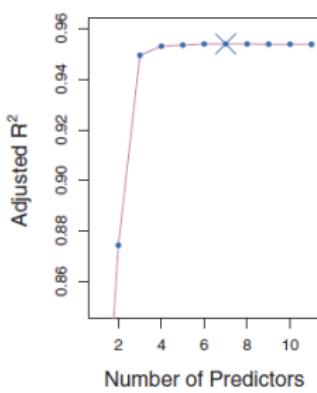
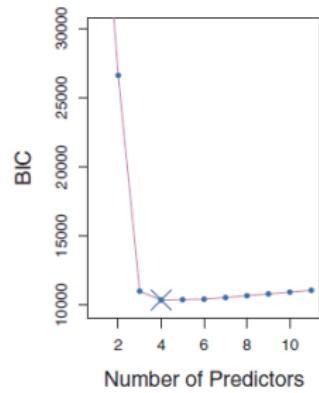
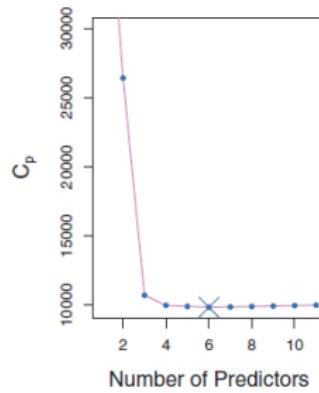
- Like FSS, BSS approach searches through $1 + p(p + 1)/2$ models and so can be applied in settings where p is too large to apply best subset selection.
- Like FSS, BSS is not guaranteed to yield the best model containing a subset of p predictors.
- BSS requires the number of observations n be larger than p so that the full model can be fit, however FSS can be used even when $n < p$ and so is the only viable subset method when p is very large.

Choosing the Optimal Model

- The model containing more predictors tend to produce smaller RSS and larger R^2 . **Why?**
- We wish to choose the model with low test error.
- Therefore, RSS and R^2 are not appropriate to be set as a criterion for selecting the best model among a collection of models with different number of predictors.
 - ① We can estimate the test error by **making adjustment to the training error** to account for the bias due to overfitting.
 - ② Or a different **cross validation** type of approaches can be applied.

C_p , AIC, BIC and Adjusted R^2

- These techniques adjust the training error for the model size and could be used to select among a set of models with different numbers of variables.
- Credit data example:



C_p and AIC

- Mallow's C_p :

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2),$$

where d is the number of parameters used and $\hat{\sigma}^2$ is an estimate of the variance of the random error. C_p is an unbiased estimate of test MSE.

- The **AIC** is defined for a large class of models fit by maximum likelihood:

$$AIC = -2 \log L + 2d,$$

where L is the maximized value of the likelihood function of the estimated model

- Linear models with Gaussian noise, C_p and AIC are equivalent.

- The **BIC** is defined as

$$BIC = -2 \log L + \log(n)d,$$

or i.e. for linear model with Gaussian noise

$$BIC = \frac{1}{n}(RSS + \log(n)d\hat{\sigma}^2).$$

- BIC takes on a smaller value for a model with lower test error, so we tend to select the model with the lowest BIC value.
- Since $\log(n) > 2$ for any $n > 7$, the BIC penalty generally places a heavier penalty on models with many predictors. **which one of C_p and BIC tends to select models with smaller number of predictors?**

Adjusted R^2

- For a least square model with d predictors, the adjusted R^2 is defined as

$$\text{Adjusted } R^2 = 1 - \frac{\text{RSS}/(n - d - 1)}{\text{TSS}/(n - 1)},$$

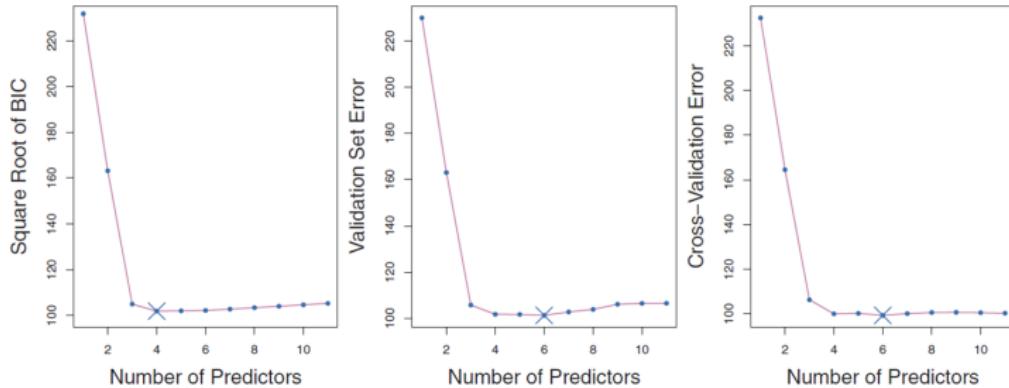
where TSS denotes the total sum of squares. Unlike C_p , AIC, BIC, a larger values of adjusted R^2 reveals a model with smaller test error.

- Maximizing the adjusted R^2 is equivalent to minimizing $\text{RSS}/(n - d - 1)$, which may increase or decrease (**Why?**), due to the presence of d in the denominator.
- Adjusted R^2 pays a price for including irrelevant predictors in the model.

Cross Validation

- All approaches generate model \mathcal{M}_k for $k = 0, 1, \dots$, we aim to select an optimal k .
- We can compute the **cross validation error** for each model \mathcal{M}_k and choose the one with the smallest test error.
- Both provide a **direct estimate** of the test error and makes **fewer assumption** about the true model.
- It can also be used in a wider range of model selection tasks, even in cases where it is hard to pinpoint the model degree of freedom or challenging to estimate the error variance σ^2 especially in high dimensional large p small n scenario.

Credit Data Example



- Validation set: 75% as training set and 25% as validation set.
- 10 fold CV approach: both CV methods select six variables.
- **One-standard-error rule:** Calculate the standard errors of the estimated test MSE for each model size, and select the smallest model for which the estimated test error is within one standard error of the lowest point of that curve. **Credit data example?**

Ridge Regression

The Lasso and its Variants

Ridge Regression

- Shrinkage methods fit the model containing all p predictors using the approaches that regularizes the coefficient estimates, or i.e. shrinks the estimated coefficients towards zero.
- Ridge regression is motivated from removing multicollinearity effect.
See the white board.
- Shrinkages method are developed to reduce the large variance.

Ridge Regression

- The least square estimate $\beta_0, \beta_1, \dots, \beta_p$ through minimizing

$$RSS = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2.$$

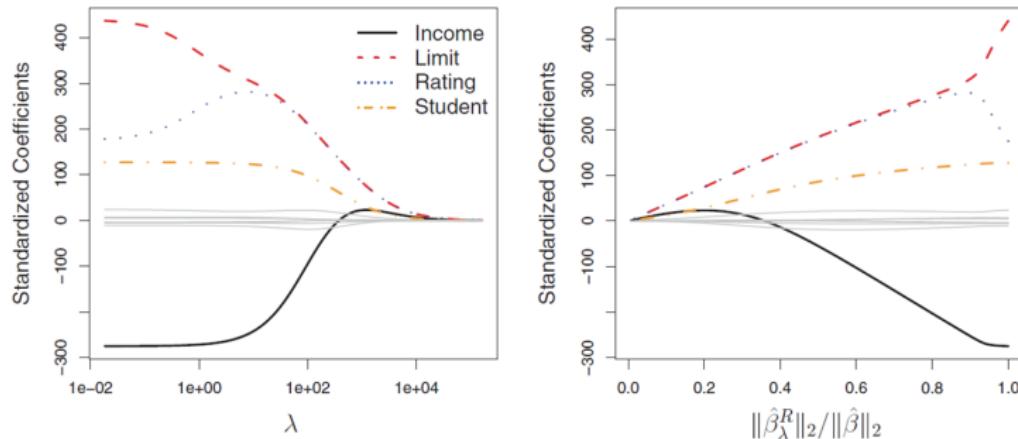
- Ridge regression estimates $\hat{\beta}_\lambda^R$ through minimizing

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = RSS + \text{Penalty},$$

where $\lambda \geq 0$ is the regularization parameter, to be determined separately.

- Larger or smaller λ ? Selecting a good λ is very important, cross validation or AIC/BIC can be used for this (to be discussed).
- Note the shrinkage penalty is applied to the slope **not the intercept**.

Credit Data Example (Solution Paths)



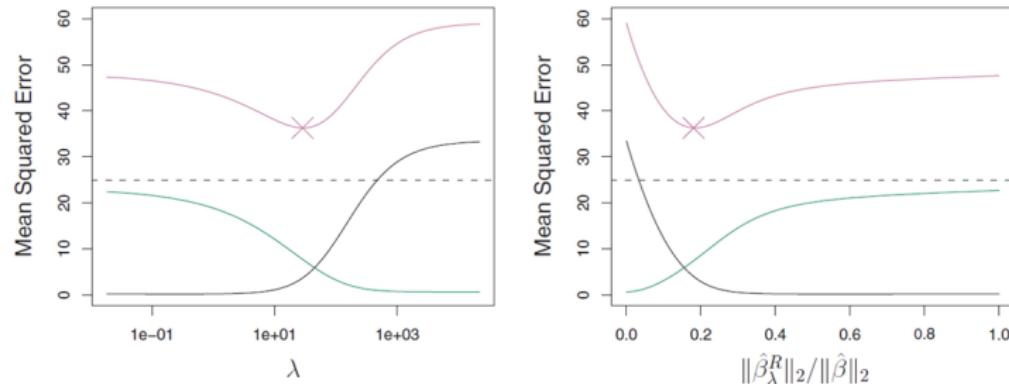
- Left: each curve corresponds to the ridge regression coefficient estimate for one of ten variables.
- Right: instead of plotting vs the regularization parameter, we display $\|\hat{\beta}_\lambda^R\|_2 / \|\hat{\beta}\|_2$, where $\hat{\beta}$ denotes the LSE.

Scaling of Predictors in Ridge Regression

- The standard LSE are **scale invariant**, multiplying X_j by a constant c will lead to the LSE by a factor of $1/c$. So $X_j \hat{\beta}_j$ remain the same.
- Ridge regression coefficient estimates can change significantly when a predictor is multiplied by a constant. (**why? some mathematics.**)
- Hence it is best to perform **standardization for the predictors** before applying ridge regression,

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}}.$$

Bias and Variance Tradeoff for Ridge Regression



Simulated data with $n = 50$ observations and $p = 45$ predictors all having nonzero coefficients. Squared bias (black), variance (green) and test MSE (purple) for the ridge regression predictions on a simulated data set. The horizontal dashed lines indicate the minimum possible MSE.

The Lasso

- Ridge regression still include all p predictors in the model. (No variable selection!)
- The lasso (Robert Tibshirani, 1996, Journal of Royal Statistical Society: Series B) was proposed to overcome this advantage. **The first inventor?**
- The lasso coefficient $\hat{\beta}_\lambda^L$

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j| = RSS + \text{Penalty},$$

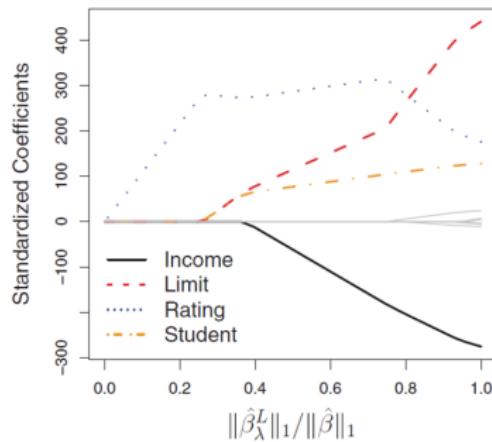
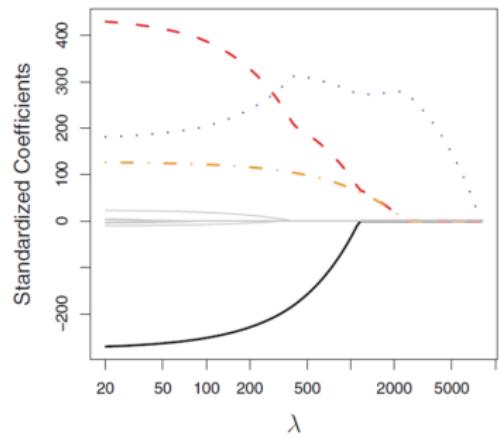
where $\lambda \geq 0$ is the regularization parameter.

- Matrix representation? See the homework.
- Ridge regression considers ℓ_2 norm for the penalty term and the lasso uses a ℓ_1 norm of penalty.

The Lasso

- The lasso also shrinks the coefficient estimates towards zero.
- However, ℓ_1 penalty has the effect of forcing some of the coefficients to be exactly zero where a large λ is used.
- The lasso can perform **variable selection** and choose a **sparse** model containing a subset of all p predictors.
- What about the selection of regularization parameter? CV or AIC/BIC.

Credit Data Example (Solution Paths)



Another Formulation for Ridge Regression and the Lasso

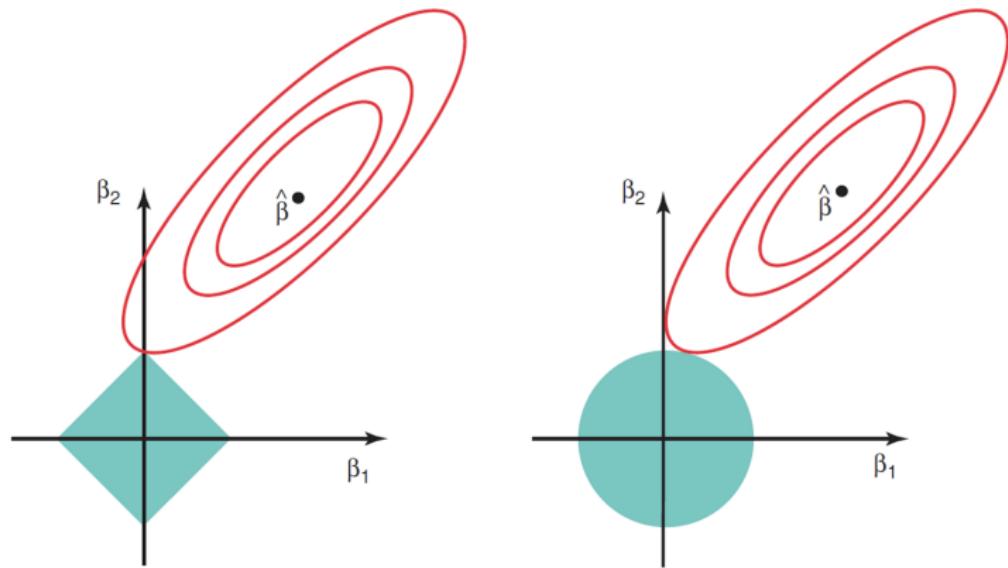
- One can easily show that the lasso and ridge regression coefficient estimates solve the following problems

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad \text{subject to } \sum_{j=1}^p |\beta_j| \leq s$$

and

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad \text{subject to } \sum_{j=1}^p \beta_j^2 \leq s.$$

A Graphical Illustration

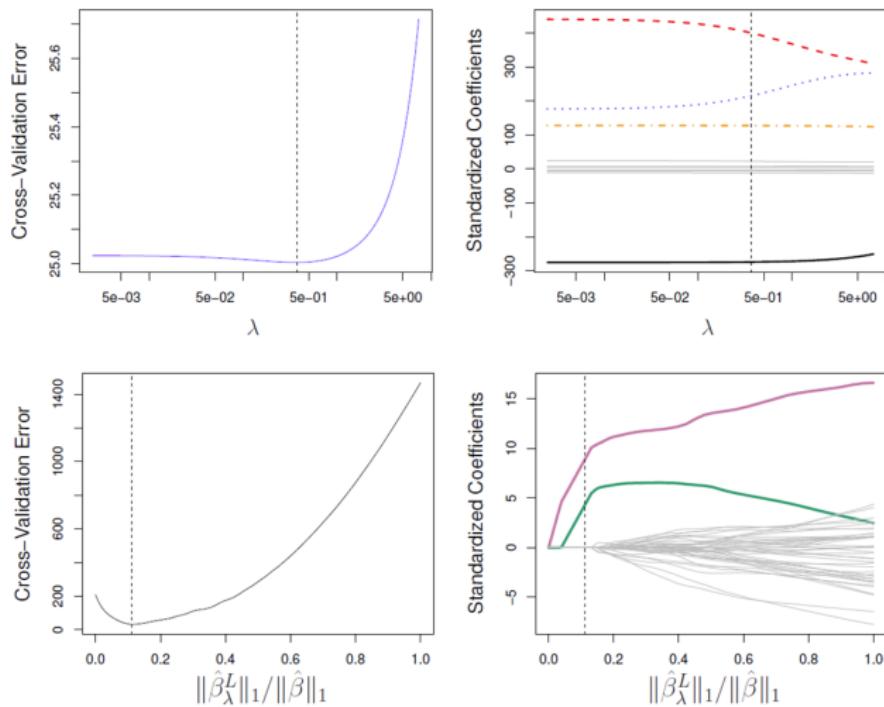


The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.

Tunning Parameter Selection for Ridge Regressiona and the Lasso

- **Bias and variance tradeoff:** As λ increase, the model becomes sparser, so that the variance will decrease and the bias will increase. The test MSE will behave like a U shape.
- CV methods provide a simple way to determine which of the models with different sparsity levels is the best. We choose a grid of λ values and compute the CV errors for each λ .
- We then select the regularization parameter for which CV error is the smallest.
- Finally, the model is re-fitted using all of the available observations and the selected values of the regularization parameter. **Why refit? downside of the lasso.**

Credit Data Example (Simulation 2)



A Simple Special Case for Ridge Regression and the Lasso

- Consider a simple special case $n = p$ and the design matrix \mathbf{X} is an identity matrix. Also consider the regression without an intercept.
- Least squares** aim to minimize $\sum_{j=1}^p (y_j - \beta_j)^2$ and the solution is given by

$$\hat{\beta}_j = y_j, j = 1, \dots, p.$$

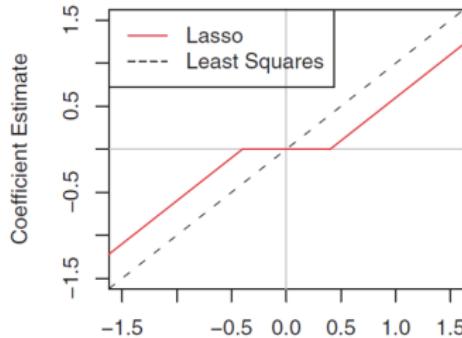
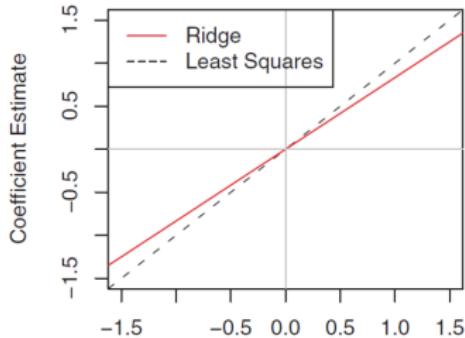
- Ridge regression** considers minimizing $\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2$ and the solution is given by

$$\hat{\beta}_j^R = y_j / (1 + \lambda)$$

- The lasso** considers minimizing $\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$ and the solution is given by

$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2 \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2 \\ 0 & \text{if } |y_j| \leq \lambda/2 \end{cases}$$

Graph Illustration



- Ridge regression more or less shrinks every dimension of the data by the same proportion.
- The lasso more or less shrinks all coefficients towards zero by a similar amount and sufficiently small coefficients are shrunken all the way to zero. (**soft thresholding**). Any **hard thresholding**? See next page, the whiteboard and the homework.

Best Subset Selection

- Like the constraint versions of ridge regression and the lasso, the best subset selection can be thought of the following constraint problem

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad \text{subject to } \sum_{j=1}^p I(\beta_j \neq 0) \leq s.$$

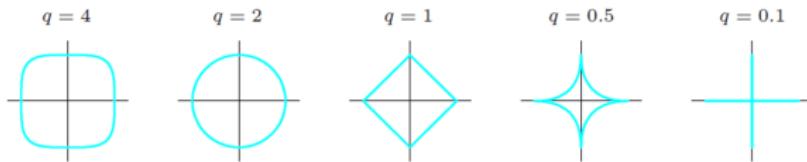
- ℓ_0 type of constraint.
- What is the unconstrained version of the optimization problem?
- ℓ_0 penalty.

Generalize Ridge Regression and the Lasso

- Consider the penalized criterion

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|^q$$

- Contours of constant values of the $\sum_j |\beta_j|^q$ for given values of q



- The case ($q = 1$) (lasso) is the **smallest** q such that the constraint region is **convex**; non-convex constraint regions make the optimization problem more difficult! (What about $q > 1$?)

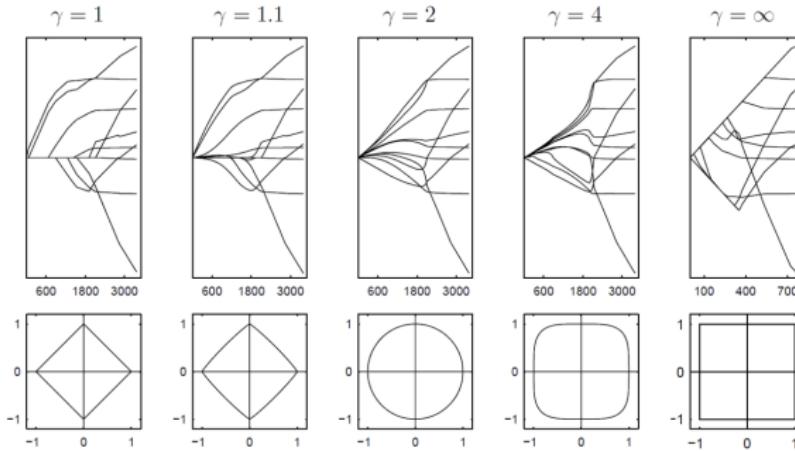
Bridge Regression (Frank and Friedman, 1993)

- Consider the penalized optimization problem

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|^q,$$

where $q \geq 1$.

- We call the solution **bridge estimator**. (Is this what we want?)



ℓ_0 , ℓ_1 and ℓ_2 Penalty Terms

- The lasso is the only one that provides a **sparse** solution (variable selection) and also a **convex** optimization problem.
- **The lasso:** ℓ_1 penalty, shrinks all coefficients toward zero by a similar amount (**Biased estimators for large coefficients!**).
- **Best subset selection:** ℓ_0 penalty (hard thresholding), but not continuous(**not stable in model prediction!**).
- **Ridge regression:** ℓ_2 penalty (**no sparsity!**).
- Hence, **good** penalty functions should result in an estimator with three properties
 - ① Unbiasedness
 - ② Sparsity
 - ③ Continuity
- What about ℓ_0 , ℓ_1 and ℓ_2 penalties?
- Combine some penalties?

Group Lasso Penalty

- Suppose p predictors are divided into G groups, with p_g the size in group g , $g = 1, \dots, G$. The group lasso minimizes the following convex criterion

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{g=1}^G \mathbf{x}_{ig}^T \boldsymbol{\beta}_g \right)^2 + \lambda \sum_{g=1}^G \|\boldsymbol{\beta}_g\|_2,$$

where $\mathbf{x}_{ig} \in \mathbb{R}^{p_g}$, $i = 1, \dots, n$ denote the predictors within g -th group with the corresponding coefficient vector $\boldsymbol{\beta}_g \in \mathbb{R}^{p_g}$.

- Since ℓ_2 norm of a vector is zero if and only if all its components are zero, this procedure encourages sparsity at both group and individual levels.
- Group lasso penalty enforces **group sparsity** (all within-group components are zero.)

The Penalized Likelihood Approach

- Rather than considering minimizing penalized least squares for linear model with Gaussian noise, we consider more general case.
- A **general penalized likelihood approach** considers minimizing

Negative log likelihood + $\lambda \ell_1$ norm,

where λ is a non-negative parameter to tune the sparsity.

Principal Component Analysis (PCA)

Principal Component Regression (PCR)

Dimension Reduction Methods

- Least squares or shrinkage approaches consider fitting linear regression models using the original predictors.
- We now explore a class of approaches that **transform** the predictors and then fit a least squares model using the transformed variables. We will refer to these techniques as *dimension reduction methods*.

Dimension Reduction Methods

- Least squares or shrinkage approaches consider fitting linear regression models using the original predictors.
- We now explore a class of approaches that **transform** the predictors and then fit a least squares model using the transformed variables. We will refer to these techniques as *dimension reduction methods*.
- Let Z_1, Z_2, \dots, Z_M represent $M < p$ linear combinations of our original p predictors X_1, X_2, \dots, X_p . That is

$$Z_m = \sum_{j=1}^p \phi_{mj} X_j$$

for some constants $\phi_{m1}, \dots, \phi_{mp}$, $m = 1, \dots, M$.

- We can then fit the linear regression model

$$y_i = \theta_0 + \sum_{m=1}^M \theta_m z_{im} + \varepsilon_i, i = 1, \dots, n.$$

- Note the regression coefficients in the above linear models are given by $\theta_0, \theta_1, \dots, \theta_M$, we need to find a way to choose the constants $\phi_{m1}, \dots, \phi_{mp}$.

Dimension Reduction Methods in Details

- The term **dimension reduction** comes from the fact that it reduces the problem of estimating $p + 1$ coefficients to $M + 1$ coefficients, $M < p$.
- We have

$$\sum_{m=1}^M \theta_m z_{im} = \sum_{m=1}^M \theta_m \sum_{j=1}^p \phi_{mj} x_{ij} = \sum_{j=1}^p \sum_{m=1}^M \theta_m \phi_{mj} x_{ij} = \sum_{j=1}^p \beta_j x_{ij},$$

where $\beta_j = \sum_{m=1}^M \theta_m \phi_{mj}$.

- Hence the reduced model can be thought a special case of the original model and dimension reduction serves to constrain the estimated β_j coefficients.
- What about the bias and variance tradeoff analysis for M compared with p ?

Two Steps in Dimension Reduction Methods

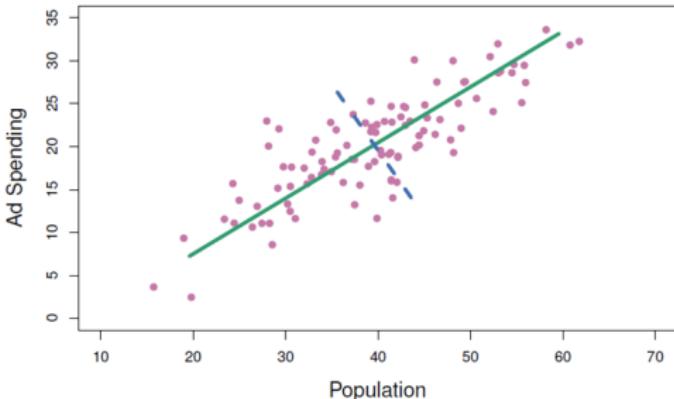
- ① The transformed predictors Z_1, \dots, Z_M are obtained.
- ② The model is fit using these M predictors

However, the choice of Z_1, \dots, Z_m , or equivalently, the selection of $\phi_{m1}, \dots, \phi_{mp}$ can be achieved in different ways. We consider the approach **principal components regression**.

Principal Components Regression

- Here we use **principal component analysis (PCA)** (with details discussed in Week 11) to define the linear combinations of the predictors in our regression setting. PCA is a technique for reducing the dimension of $\mathbf{X} \in \mathbb{R}^{n \times p}$.
 - ① The first PC is that (normalized) linear combinations of the variables with the largest variance.
 - ② The second PC has the largest variance subject to being uncorrelated to the first one.
 - ③ So on...
- We substitute the original correlated variables with **a small set of principal components** which capture the joint variation.
- Here $\phi_{m1}, \dots, \phi_{mp}$ are the **principal component loadings** and Z_1, \dots, Z_M are the **principal component scores**.

Graphical Illustration of PCA

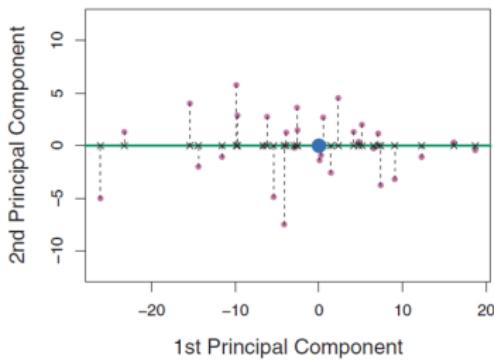
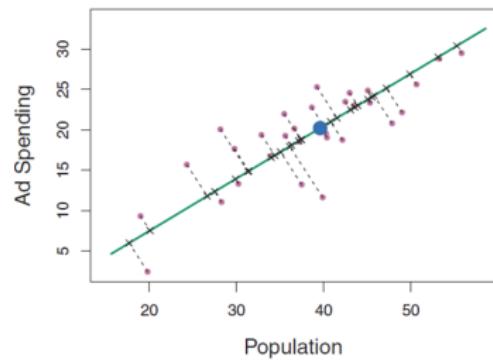


The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.

The first PC: we project the 100 observations onto the line which would lead to the largest possible variance.

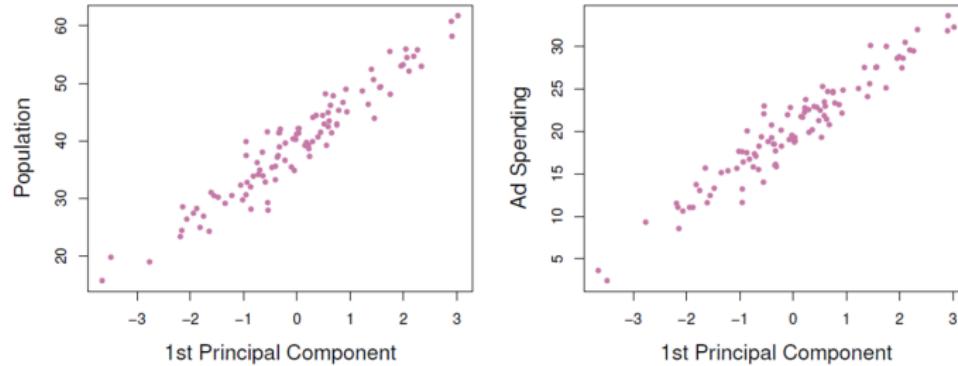
More on PCA

Another interpretation: The first PC vector defined the line that is **as close as possible to the data.** (Projected observations are as close as possible to the observations.)



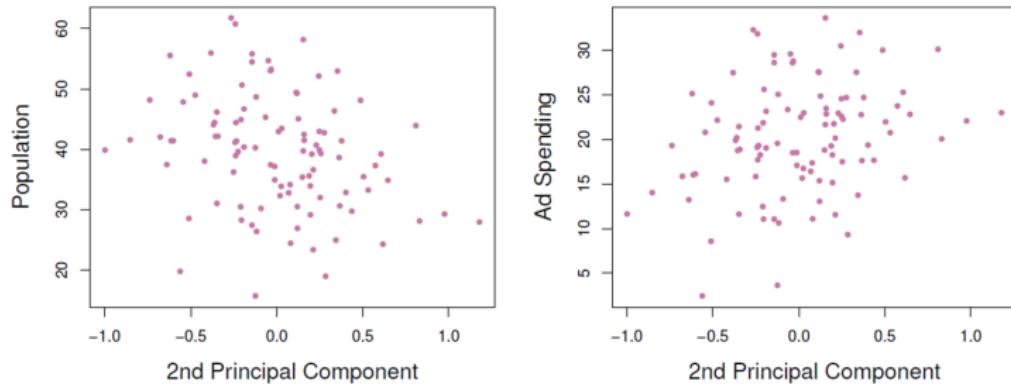
The second PC vector is perpendicular to the first PC direction

Plot of PC scores



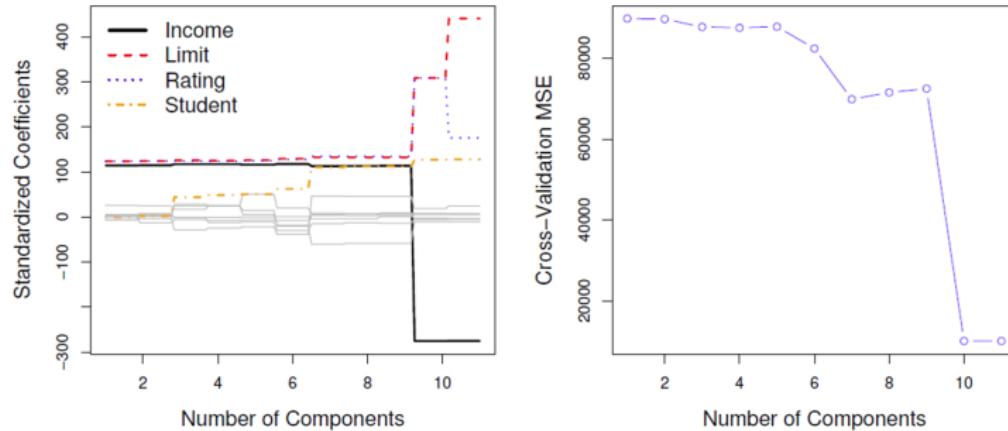
Plots of the first PC scores $z_{i1}, i = 1, \dots, n$, versus variables *population* and *ad spending*, respectively. The relationships are strong.

Plot of PC scores



Plots of the second PC scores $z_{i2}, i = 1, \dots, n$, versus variables *population* and *ad spending*, respectively. The relationships are weak.

Choosing the Number of Reduced Dimension M ?



Left: PCR standardized coefficients estimates on the Credit dataset for different values of M . Right: The plot of 10-fold cross validated MSE obtained using PCR versus M .

ST 443: Machine Learning and Data Mining

Dr. Xinghao Qiao

Columbia House, Room 5.15

x.qiao@lse.ac.uk

Department of Statistics



Office Hours: Tuesday 4:30–5:30pm

Lecture 5

Q1. Polynomial Regression

Q2. Regression Splines

Q3. Smoothing Splines

Q4. Generalized Additive Models (GAM)

Moving Beyond Linearity

- So far in the course we have only discussed linear models. Linear models have significant advantages in terms of interpretation and inference. However, standard linear regression has significant limitation in terms of prediction powers.
- We have seen how to improve OLS using the ridge regression and the lasso, but we are still using a linear model.
- In this lecture, we will start to relax the linearity assumption while still attempting to maintain as much of the interpretability as possible.
- We will do this by examining very simple extensions like **polynomial regression** as well as more sophisticated approaches such as **regression splines**, **smoothing splines** and **GAM**.

Polynomial Regression

Polynomial Regression

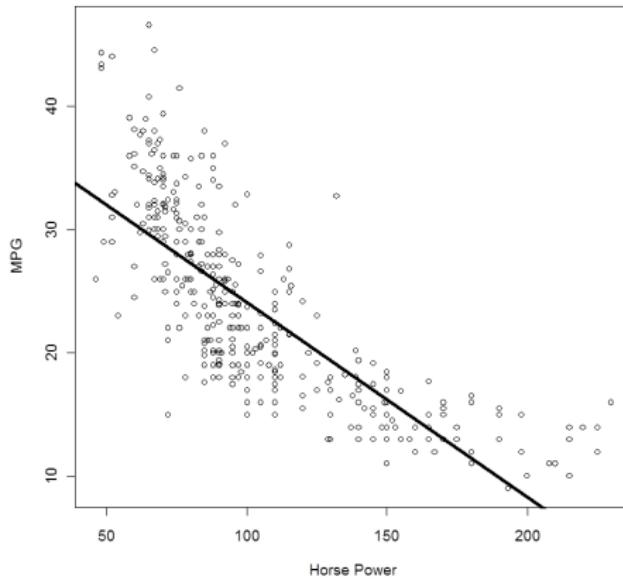
Step Functions

Basis Functions

MPG vs Horsepower

- Linear regression model:

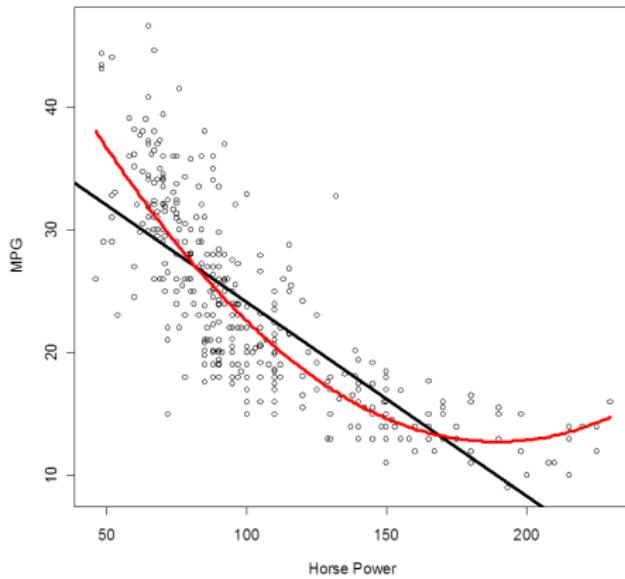
$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, i = 1, \dots, n.$$



MPG vs Horsepower

- Quadractic regression model:

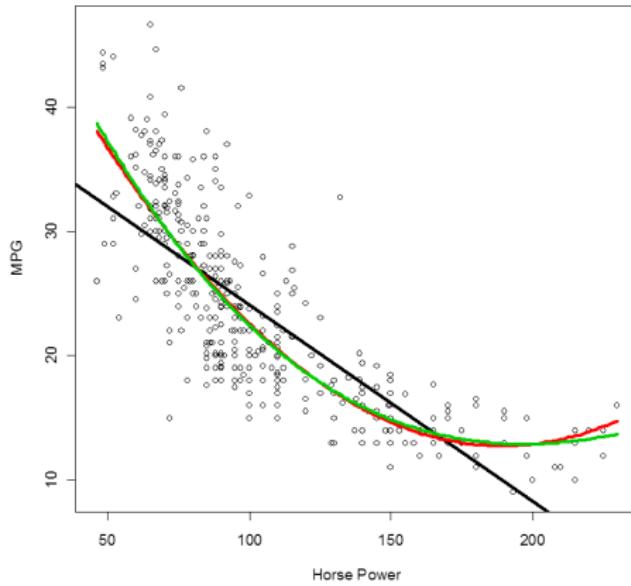
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i, i = 1, \dots, n.$$



MPG vs Horsepower

- Cubic regression model:

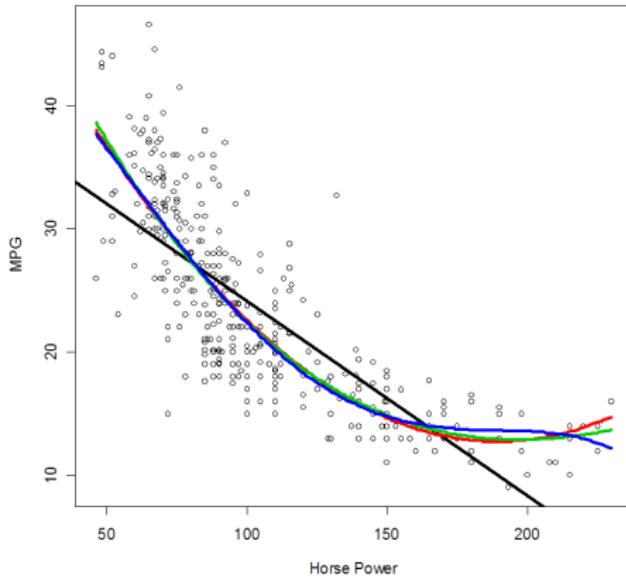
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \varepsilon_i, i = 1, \dots, n.$$



MPG vs Horsepower

- X raised to the power of 4:

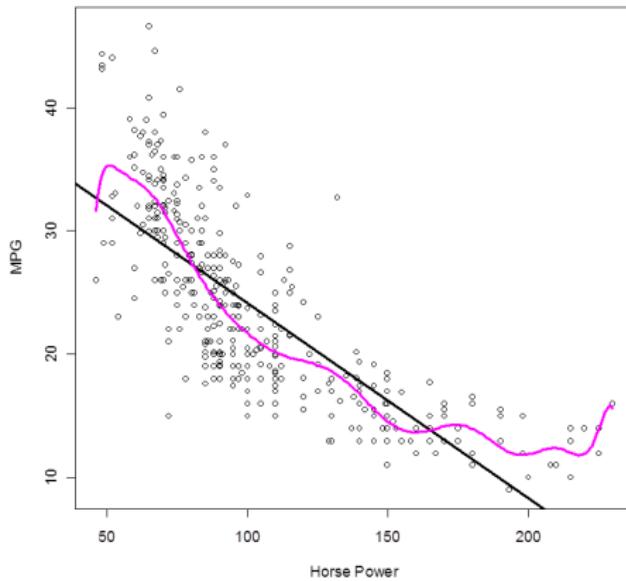
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \varepsilon_i, i = 1, \dots, n.$$



MPG vs Horsepower

- X raised to the power of 14:

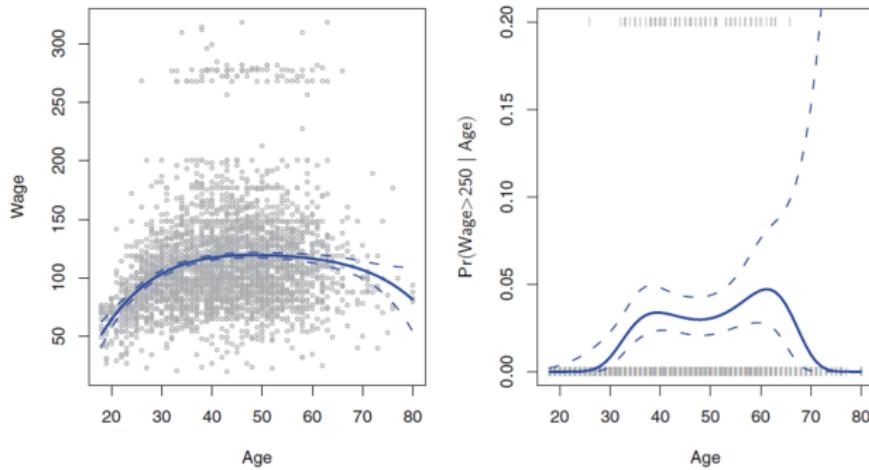
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_{14} x_i^{14} + \varepsilon_i, i = 1, \dots, n.$$



Polynomial Regression

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d + \varepsilon_i, i = 1, \dots, n.$$

Degree-4 Polynomial



The **Wage** data: the **solid blue curve** is a degree-4 polynomial of **wage** (in thousands of dollars) as a function of **age**, fit by least squares.

Details

- Not really interested in the coefficients; more interested in the fitted function values at any x_0 :

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4.$$

- Since $\hat{f}(x_0)$ is a linear function of the $\hat{\beta}_k$'s, we can derive a simple expression for pointwise variance $\text{Var}[\hat{f}(x_0)]$ at any value x_0 .
- In the figure, we show $\hat{f}(x_0) \pm 2\text{se}[\hat{f}(x_0)]$.
- The degree d can be chosen using cross-validation approaches treating it as a tuning parameter.

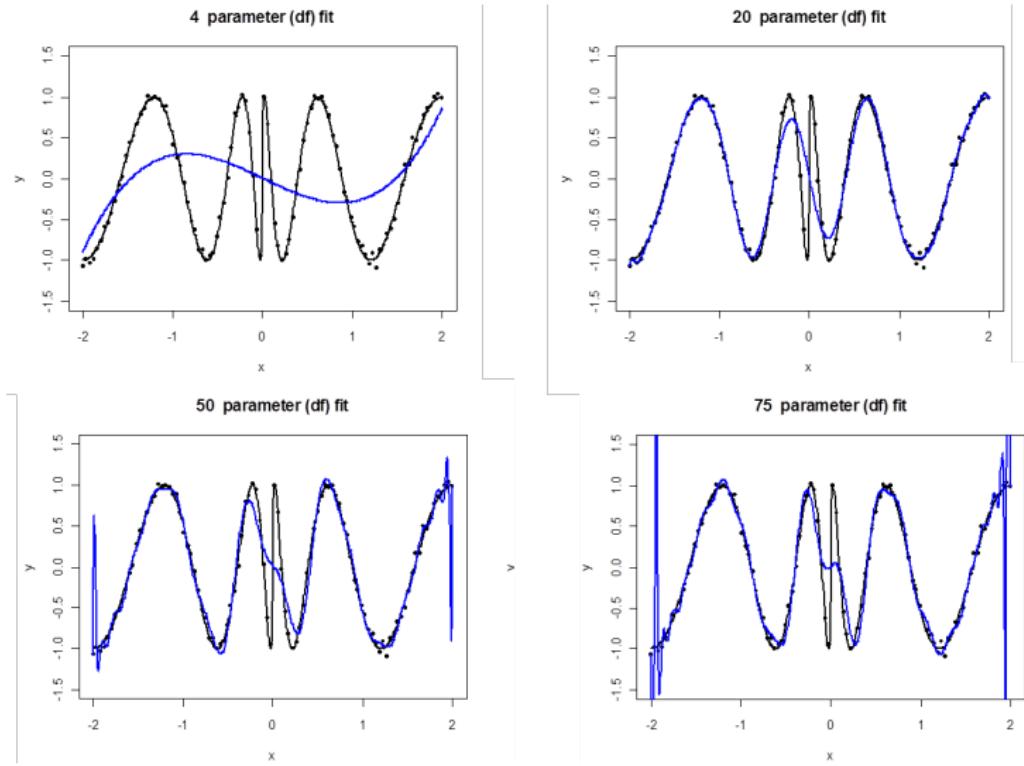
More on Polynomial Regression

- For the purpose of classification, logistic regression (right-hand panel of the figure) can be used

$$P(y_i > 250 | x_i) = \frac{\exp(x_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d)}{1 + \exp(x_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d)}.$$

- To obtain the confidence intervals shown in the figure, we first compute the upper and lower bound on the **logit scale** and then convert to get on the **probability scale**.
- In R, we use $y \sim \text{poly}(x, \text{degree} = 3)$.
- Caveat: notorious tail behaviour.

A Hard Simulation Example



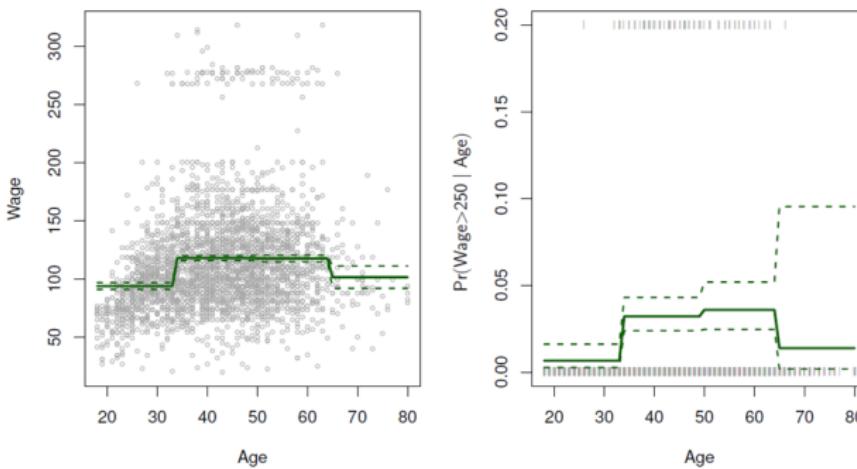
Step Functions

- Another way of performing transformation of a variable creates the range of X into bins

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \cdots + \beta_K C_K(x_i) + \varepsilon_i,$$

where $C_0(x_i) = I(x_i < c_1)$, $C_k(x_i) = I(c_k \leq x_i \leq c_{k+1})$ for $k = 1, \dots, K - 1$ and $C_K(x_i) = I(c_K \leq x_i)$.

Piecewise Constant



More on Step Functions

- Create a series of dummy variables to represent each group.
- Useful way of generating interaction terms that are more easily interpretable, e.g. interaction effect of `Year` and `Age`,
 $I(\text{Year} < 2005) \times \text{Age}$, $I(\text{Year} \geq 2005) \times \text{Age}$.
- In R, we can use $I(\text{Year} < 2005)$ or `cut(age, c(18, 25, 40, 65, 90))`.
- How to choose the knots (cut points) is problematic.
- Logistic regression (see the right-panel of the previous figure)

$$P(y_i > 250 | x_i) = \frac{\exp(x_0 + \beta_1 C_1(x_i) + \cdots + \beta_K C_K(x_i))}{1 + \exp(x_0 + \beta_1 C_1(x_i) + \cdots + \beta_K C_K(x_i))}.$$

Basis Functions

- Polynomial and piecewise-constant regression models are special cases of a basis function approach.
- Have at hand a family of functions of transformation that can be applied to a variable X : $b_1(X), b_2(X), \dots, b_K(X)$.
- We then fit a linear model

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_K b_K(x_i) + \varepsilon_i.$$

- The basis functions are fixed and known. What are the basis functions for the polynomial regression and piecewise constant functions?
- Alternatives include, wavelets (developed by David Donoho from Stanford Statistics) Fourier series and regression splines.

Regression Splines

Smoothing Splines

Piecewise Polynomials

- Instead of fitting a high-degree polynomial over the entire range of X , **piecewise polynomial regression** involves fitting separate low-degree polynomials over different regions of X .
- Consider fitting a cubic regression model separately over different regions, the points where the coefficients changes are called **knots**, e.g.

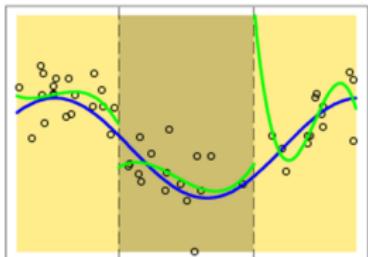
$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}^2x_i^3 + \varepsilon_i & \text{if } x_i > c \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}^2x_i^3 + \varepsilon_i & \text{if } x_i \leq c \end{cases}.$$

- K different knots throughout the range of X leads to fitting $K + 1$ different cubic polynomials.
- Constraint such as **continuity** are needed! **Why?** see next page!

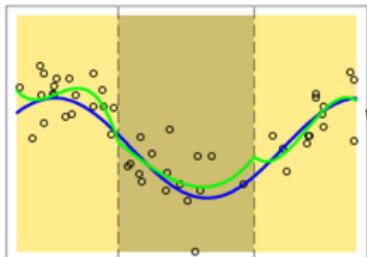
Ensuring Smoothness in the Spline Fit

1) With no constraints the curves don't even join

Discontinuous



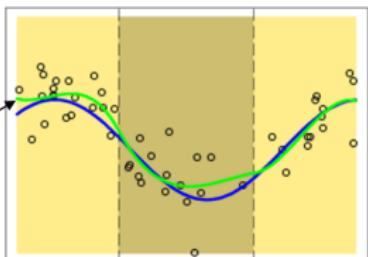
Continuous



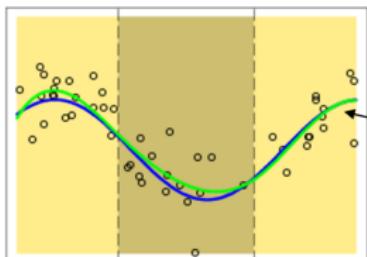
2) Now the curves join but don't look smooth

3) The curve looks a little smoother but is not quite right

Continuous First Derivative

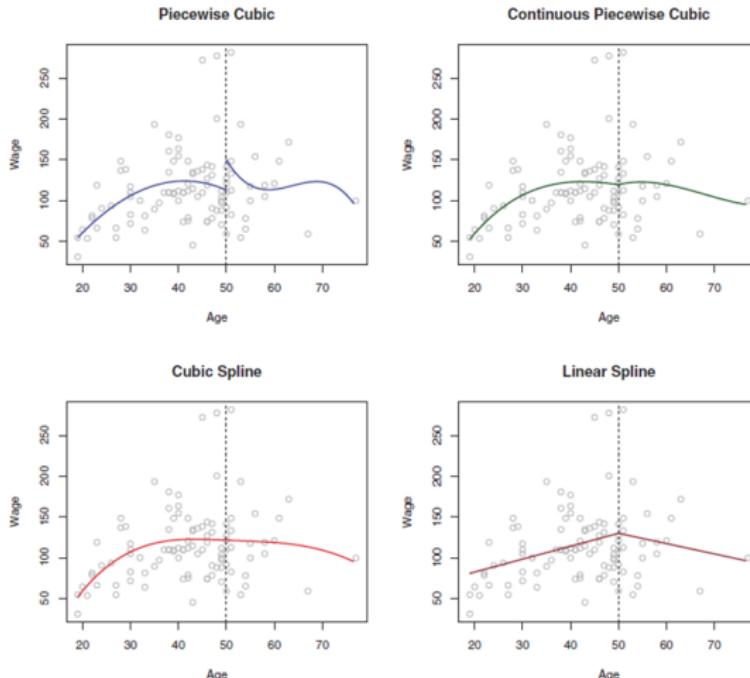


Continuous Second Derivative



4) Now we have a nice smooth looking curve.

Ensuring Smoothness in the Spline Fit



Local vs global, smooth vs non-smooth, why cubic spline is popular?
(Human eyes).

Cubic Spline

- A cubic spline with knots at $\xi_k, k = 1, \dots, K$ is a piecewise cubic polynomial with continuity up to order 2 at each knot.
- The model can be represented by

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \dots + \beta_{K+3} b_{K+3}(x_i) + \varepsilon_i,$$

where the basis functions are given by

$$b_1(x_i) = x_i$$

$$b_2(x_i) = x_i^2$$

$$b_3(x_i) = x_i^3$$

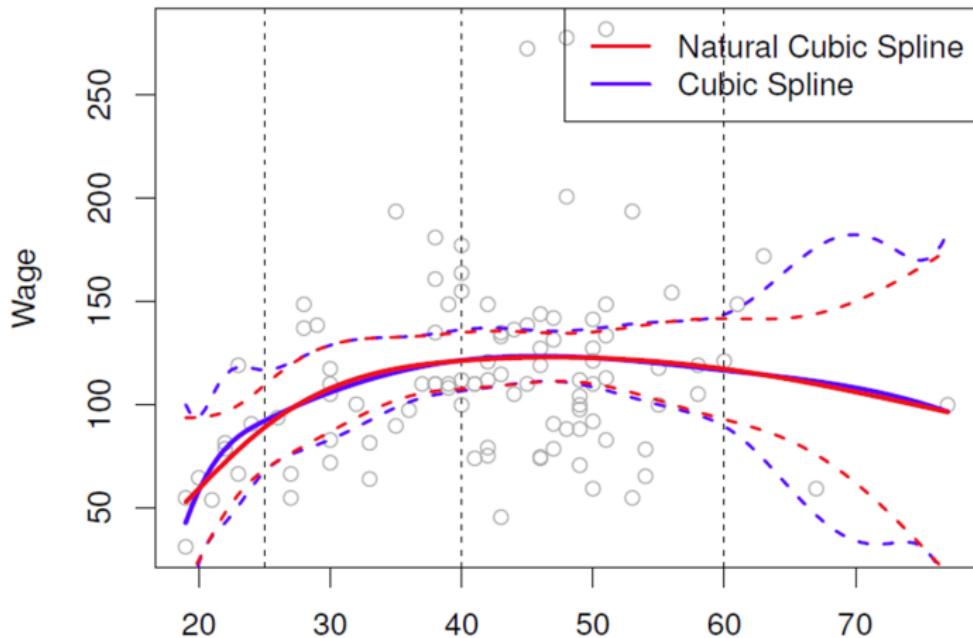
$$b_{k+3}(x_i) = (x_i - \xi_k)_+^3, k = 1, \dots, K$$

- A cubic spline with K knots uses a total of $4 + K$ degrees of freedom.
Why?
- The basis representation will lead to a discontinuity on only the third derivative at each knot.

Natural Cubic Spline

- The general definition of a degree- d spline is that it is **piecewise degree- d polynomial, with continuity in derivatives up to $d - 1$ at each knot**, e.g. linear spline, quadratic spline.
- Unfortunately, splines can have high variance at the outer range of the predictors, see the wage data example on next page.
- A **natural spline** is a regression spline with additional boundary constraints: the function is required to be **linear at the boundary** (in the region where X is smaller than the smallest knot, or larger than the largest knot). This constraint guarantees the natural splines generally produce more **stable estimates at the boundaries**.
- Why? Global tail polynomial that tails could wag quite a lot!
- A natural splines adds $4 = 2 \times 2$ extra constraints.

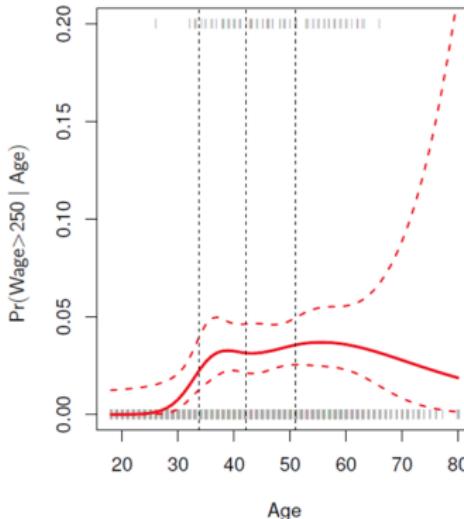
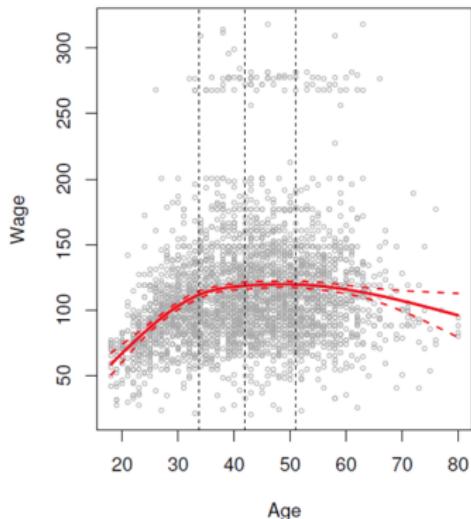
Wage Data



More on Regression Splines

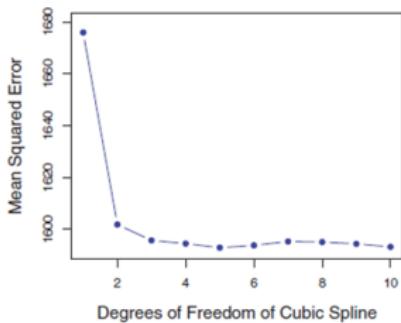
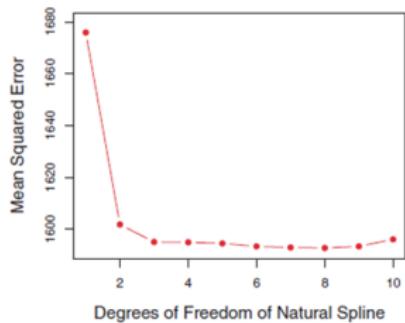
- In R: `bs(x, ...)` corresponds to regression splines and `ns(x, ...)` corresponds to natural cubic splines, both in package `splines` (will be illustrated in the computer workshop).
- Five knots including two boundary knots, what is the degrees of freedom?

Natural Cubic Spline



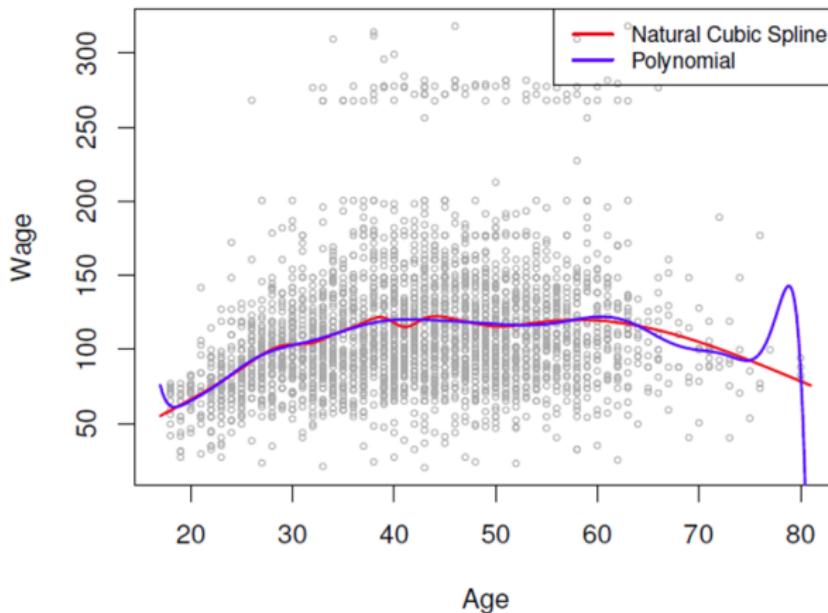
Locations and Numbers of Knots

- One strategy is to decide K , the number of knots, is to place them at appropriate quantiles for the observed X .
- A cubic spline K knots has () parameters or degrees of freedom (df).
- A natural cubic spline with K knots has () df.
- How many knots should we use? [Cross Validation](#).



Polynomial Regression vs Splines

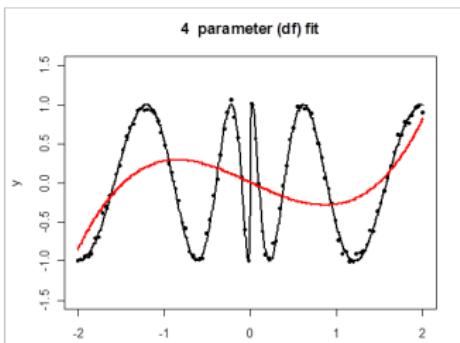
- Comparison to polynomial regression: why do the regression splines often give superior results to polynomial regression?



Regression Spline

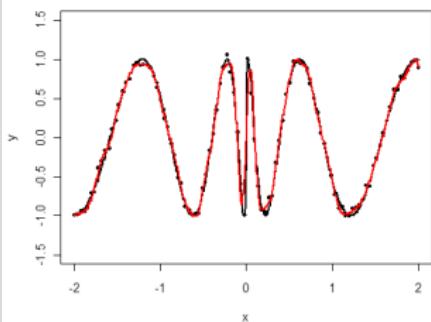
0 knots

4 parameter (df) fit

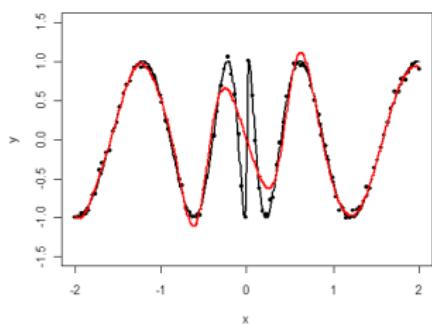


46 knots

50 parameter (df) fit

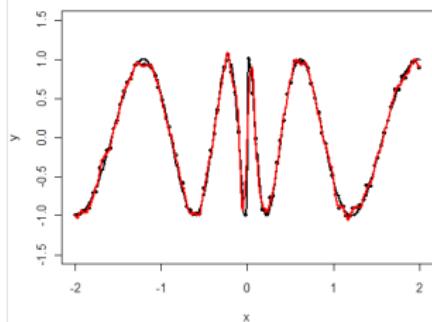


20 parameter (df) fit



16 knots

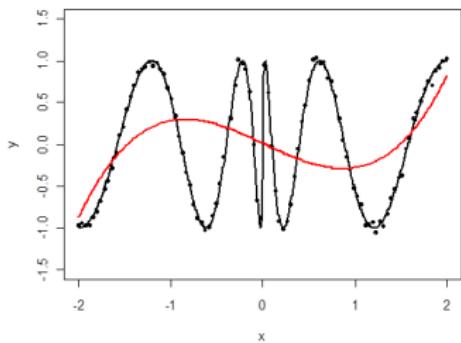
75 parameter (df) fit



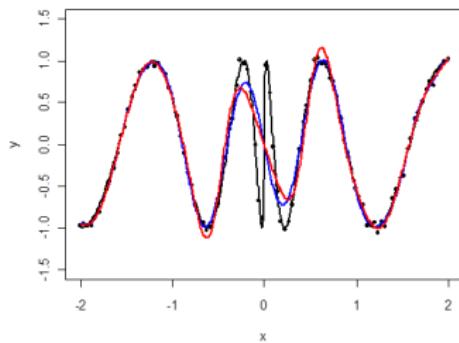
71 knots

Regression Spline vs Polynomial Regression

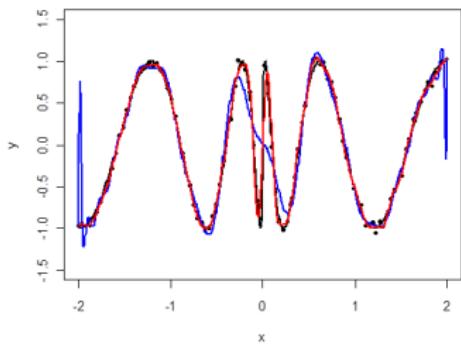
4 parameter (df) fit



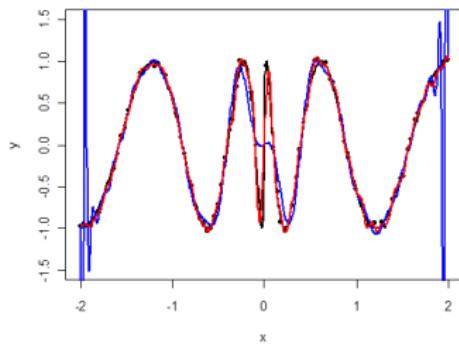
20 parameter (df) fit



50 parameter (df) fit



75 parameter (df) fit



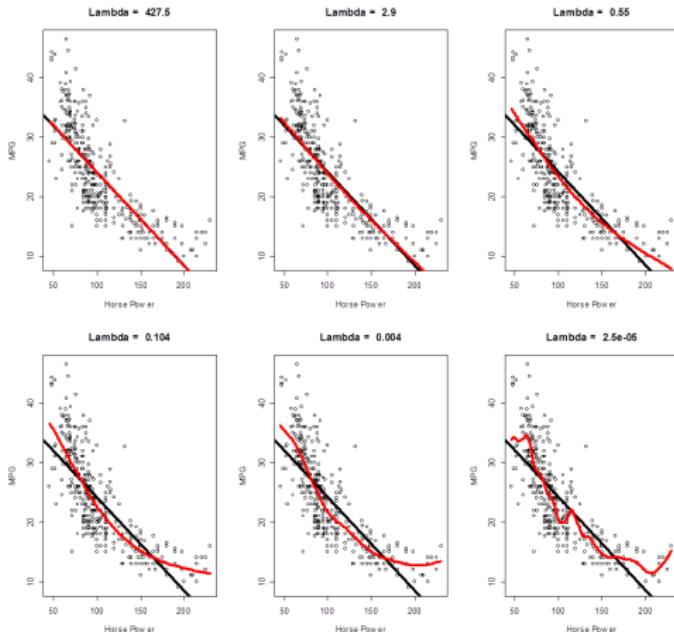
Smoothing Splines

- Consider the following criterion for fitting a smooth function $g(x)$ to some data

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

- The first term tries to make $g(x)$ match the data at each observation.
- The second term is the **roughness penalty** (1st derivative measures the slope of a function and 2nd derivative corresponds to the amount which the slope is change (*pick up wiggles of the function*)) and controls how wiggly $g(x)$ is.
- $\lambda \geq 0$ is the tuning (smoothing) parameter.
- Special cases: $\lambda = 0$ and $\lambda = \infty$?

Smoothing Parameter



- As λ approaches ∞ , $g(\cdot)$ becomes a straight line.
- As λ approaches 0, $g(\cdot)$ tries to interpolate all the points and becomes very wiggly.

More on Smoothing Splines

- The function $g(\cdot)$ that minimizes the criterion can be shown to have the properties: it is piecewise cubic polynomial with knots at unique values of x_1, \dots, x_n and continuous first and second derivatives at each knot. Furthermore, it is linear in the region outside of the extreme knots. **What is this?**
- However, it is not the natural cubic spline that we have learnt with knots at x_1, \dots, x_n -rather, it is a **shrunken version** of such a natural cubic spline, where λ controls the level of shrinkage.
- A smoothing spline will have far too many degree of freedom, but λ controls the roughness of the smoothing spline and hence the effective degree of freedom df_λ .
- Why? Although a smoothing spline contains n parameters, but these are heavily constrained or shrunken down, so df_λ is a measure of the **flexibility of the smoothing spline**.

LOOCV for Smoothing Spline

- The vector of n fitted values can be written as $\hat{\mathbf{g}}_\lambda = \mathbf{S}_\lambda \mathbf{y}$, where \mathbf{S}_λ is a $n \times n$ matrix (called smoother matrix determined by \mathbf{x}_i and λ). (How? Please see my derivation on the white board!)
- The effective degree of freedom is given by

$$df_\lambda = \text{trace}(\mathbf{S}_\lambda).$$

- Smoothing spline avoid the knot-selection issue, learning a λ to be chosen.

$$CV_{(n)}(\lambda) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{g}_\lambda^{-i}(x_i))^2 = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{g}_\lambda(x_i)}{1 - \{S_\lambda\}_{ii}} \right)^2,$$

where $\{S_\lambda\}_{ii}$ is the i -th diagonal element of \mathbf{S}_λ . This formula was introduced in Lecture 3.

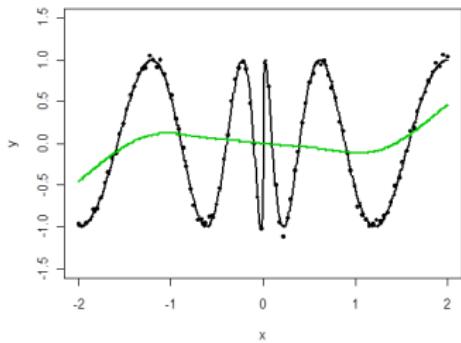
- The Generalized Cross Validation (GCV) provides an approximation to LOOCV,

$$GCV(\hat{\lambda}) = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{g}_\lambda(x_i)}{1 - \text{trace}(\mathbf{S}_\lambda)/n} \right)^2.$$

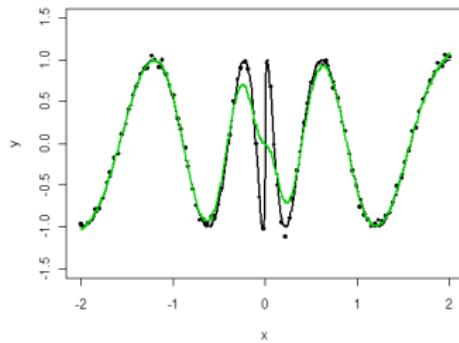
- In R, we can use `smooth.spline(age, wage, df=10)`.

Smoothing Spline

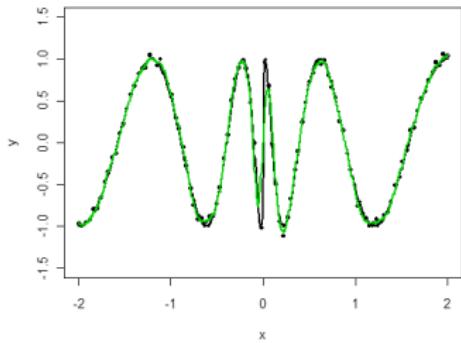
4 parameter (df) fit



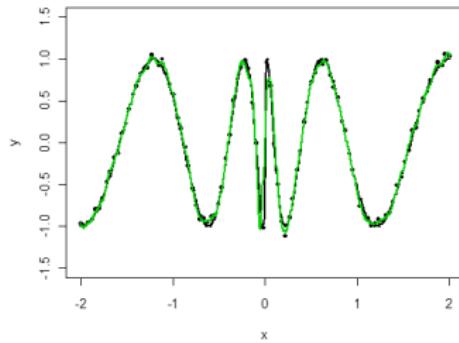
20 parameter (df) fit



50 parameter (df) fit



75 parameter (df) fit

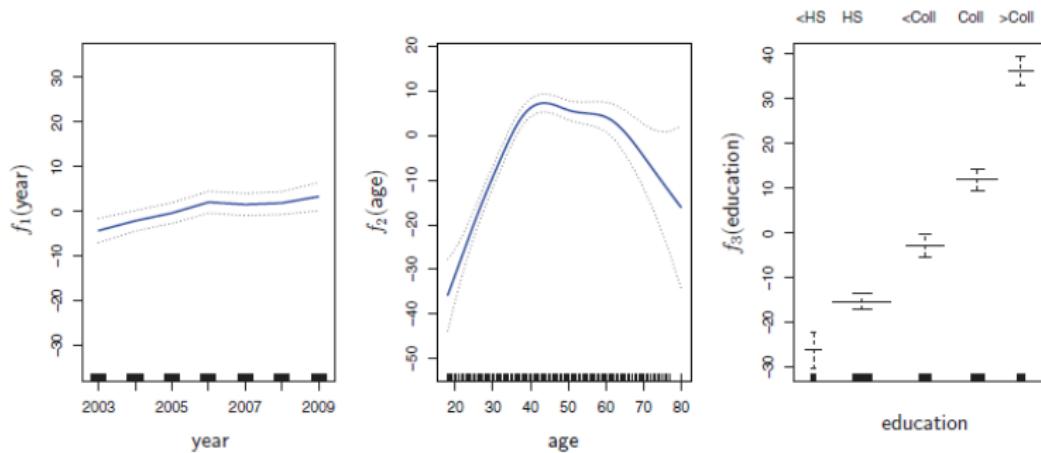


Generalized Additive Models (GAM)

- Proposed by [Hastie and Tibshirani \(Standard Statistics\)](#).
- Allows for flexible nonlinearities in several variables in the additive structure of linear models

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \varepsilon_i.$$

The wage data below



More on GAM

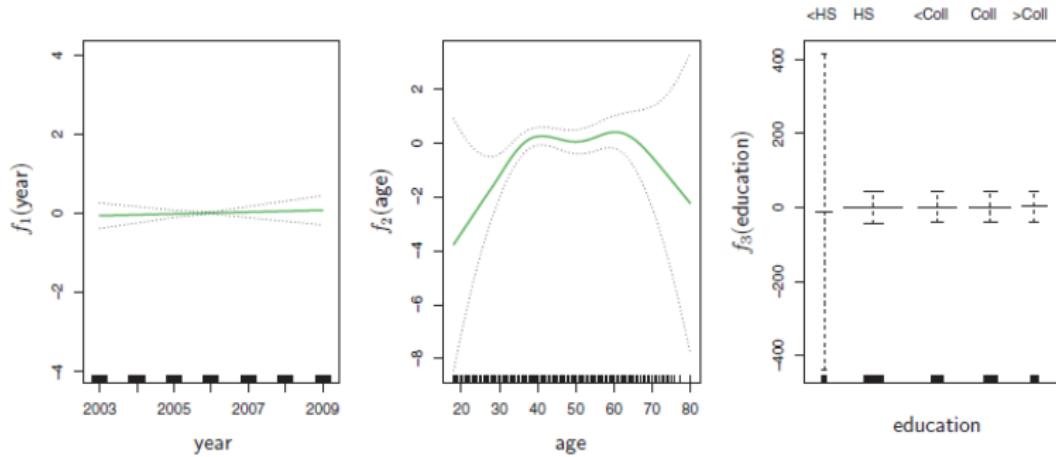
- Can fit a GAM using, e.g. natural splines:
 $lm(wage \sim ns(year, df = 5) + ns(age, df = 5) + education)$.
- Can use smoothing splines as well:
 $gam(wage \sim year + s(age, df = 5) + education)$.
- The plot can be produced using `plot.gam`.

More on GAM

- Can fit a GAM using, e.g. natural splines:
 $lm(wage \sim ns(year, df = 5) + ns(age, df = 5) + education)$.
- Can use smoothing splines as well:
 $gam(wage \sim year + s(age, df = 5) + education)$.
- The plot can be produced using `plot.gam`.
- What are pros and cons of GAMs?
- More flexible than OLS, the lasso and ridge regression, but less flexible than random forests and boosting, thus providing a useful compromise between linear and fully nonparametric models.

GAM for Logistic Regression

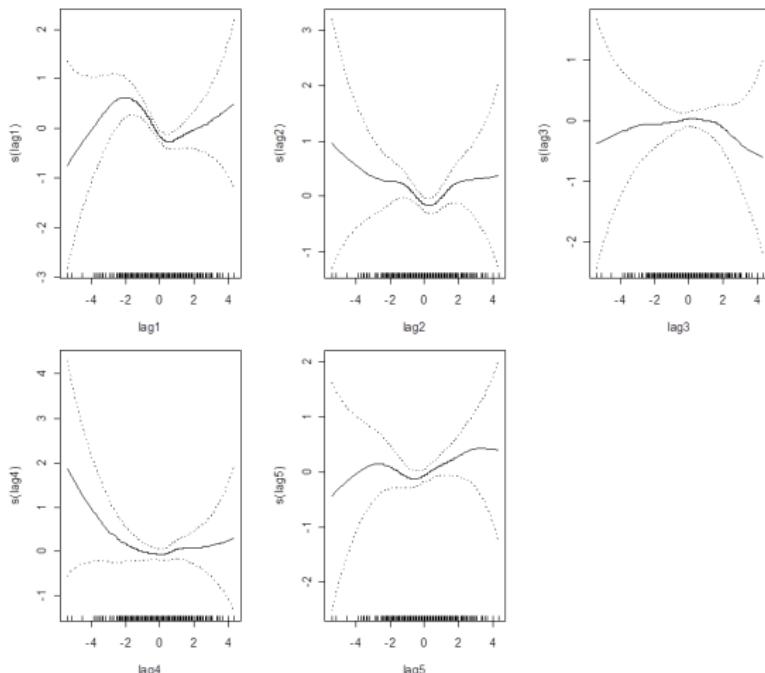
$$\log \left(\frac{p(\mathbf{x})}{1 - P(\mathbf{x})} \right) = \beta_0 + f_1(x_1) + f_2(x_2) + \dots + f_p(x_p).$$



In R: you set $\text{gam}(l(\text{wage}) > 250 \sim \text{year} + s(\text{age}, df = 5) + \text{education, family = binomial})$

GAM on S&P 500 Data

- There do seem to be some patterns here.
- In particular there seems to be evidence of a non-linear relationship between lag 1 and the probability that the market goes up today.



A Comparison

As a comparison here is the fraction of the time we are correct, using different methods, on the last 242 and using the first 1000 days to build the models.

Method	% of days correct
Guess Down	48.4%
Guess Up	51.6%
Opposite of Yesterday	52.5%
Logistic Regression	47.9%
GAM	55.8%

In Dollar Terms

- Suppose the market goes up or down 0.5% each day.
- In one year (roughly 250 trading days) if we always guess that the market will go up then our total expected return is

$$250 \times 0.516 \times 0.5\% - 250 \times (1 - 0.516) \times 0.5\% = 4\%.$$

- But if we use GAM and invest or short sell depending on its predictions our expected return is

$$250 \times 0.558 \times 0.5\% - 250 \times (1 - 0.558) \times 0.5\% = 14.5\%.$$

In Dollar Terms

- Suppose the market goes up or down 0.5% each day.
- In one year (roughly 250 trading days) if we always guess that the market will go up then our total expected return is

$$250 \times 0.516 \times 0.5\% - 250 \times (1 - 0.516) \times 0.5\% = 4\%.$$

- But if we use GAM and invest or short sell depending on its predictions our expected return is

$$250 \times 0.558 \times 0.5\% - 250 \times (1 - 0.558) \times 0.5\% = 14.5\%.$$

- Disclaimer
 - Do not sue me if you try this and lose money.
 - However, if you make some money I expect a percentage!

ST 443: Machine Learning and Data Mining

Dr. Xinghao Qiao

Columbia House, Room 5.15

x.qiao@lse.ac.uk

Department of Statistics



Office Hours: Tuesday 4:30–5:30pm

Lecture 6

Tree-Based Methods

Q1. Decision Trees

Q2. Pruning a Decision Tree

Q3. Classification Tree

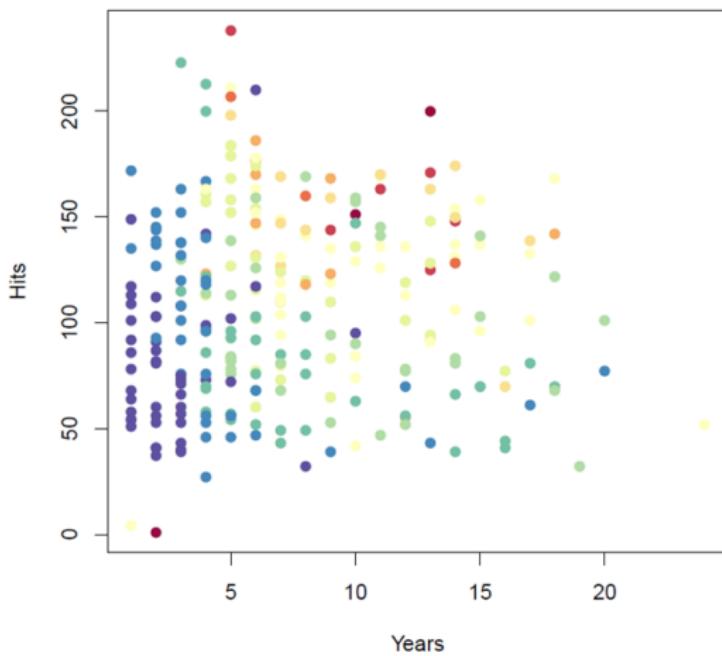
Q4. Bagging, Random Forests and Boosting

Tree-Based Methods

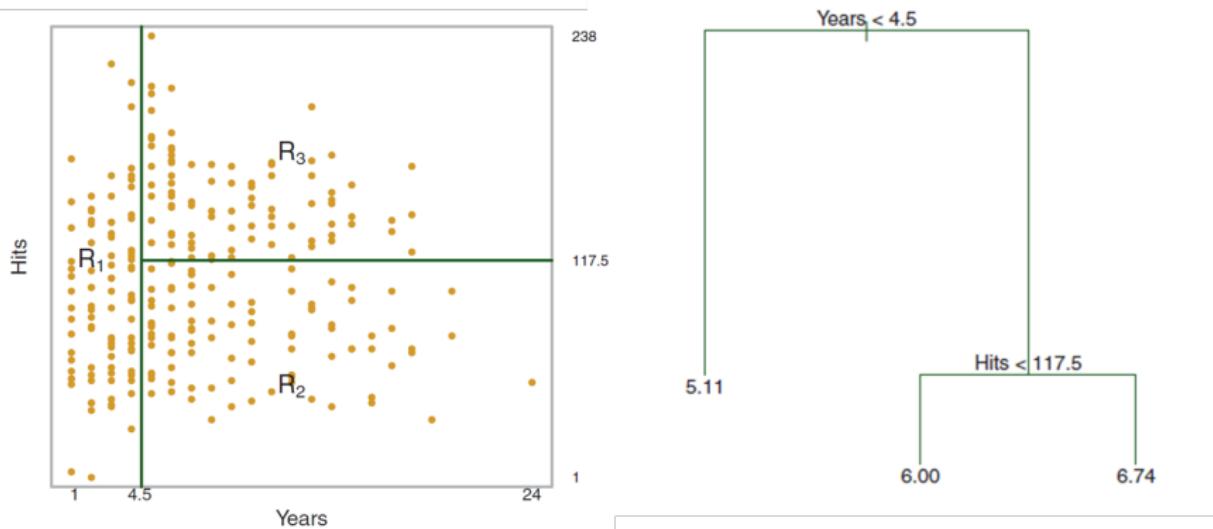
- We focus on the tree-based methods for **regression** and **classification**.
- These involve stratifying or segmenting the predictor space into a number of simple regions.
- Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these types of approaches are known as **decision tree methods**.

Baseball Player Salary Data

- Salary is colored from low (blue, green) to high (yellow, red).
- How would you stratify it?



Decision Tree for the Data



Details for the Decision Tree

- $R_1 = \{X | \text{Years} < 4.5\}$, $R_2 = \{X | \text{Years} \geq 4.5, \text{Hits} < 117.5\}$ and $R_3 = \{X | \text{Years} \geq 4.5, \text{Hits} > 117.5\}$.
- In keeping with the tree analogy, the regions R_1 , R_2 and R_3 are known as **terminal nodes** or **leaves** of the tree.
- Decision tree are typically drawn upside down, in the sense the leaves are at the bottom of the tree.
- The points along the tree where predictor space is split are referred to as **internal nodes**.
- In the hitters tree, the two internal nodes are indicated by the text $\text{Years} < 4.5$ and $\text{Hit} < 117.5$.
- We refer to the segments of the trees that connect the nodes as **branches**.

Interpretation of the Results

- **Years** is the most important factor in determining **Salary** and players with less experience earn lower salaries than more experienced players.
- Given that a player is less experienced, the number of **Hits** that he made in the previous year seems to play little role in his **Salary**.
- But among players who have been in the major leagues for five or more years, the number of **Hits** made in the previous year does affect **Salary**, and players who made more **Hits** last year tend to have higher salaries.
- The figure is likely a over-simplification, but compared to a regression model, is advantageous in displaying and interpreting.

Details in Tree Building Process

- We divide the predictor space, i.e. the set of possible values for X_1, \dots, X_p into J distinct and non-overlapping regions R_1, R_2, \dots, R_J .
- For every observation that falls into the region R_j , we make the same **prediction**, which is simply **mean of the response values for the training observation in R_j** .
- In theory, the regions could have any shape, we choose to divide the predictor space into high dimensional rectangles or boxes, for simplicity and for ease of interpretation of the resulting predictive model.
- Our target is to find boxes, R_1, \dots, R_J that minimize the RSS

$$\sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

where \hat{y}_{R_j} is the mean response for the training observation within j -th box.

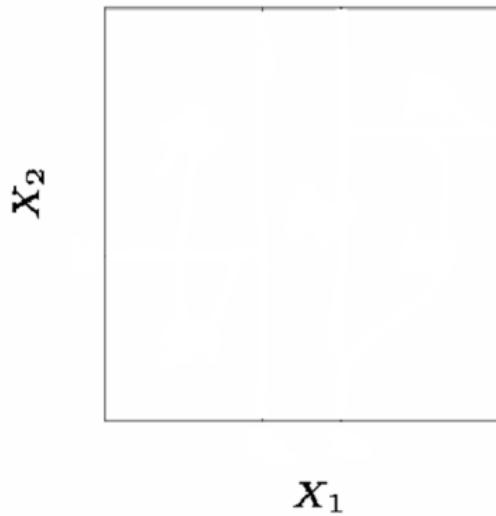
More Details in Tree Building Process

- Unfortunately, it is computational infeasible to consider every possible partition of the feature space into J boxes.
- We take a **top-down, greedy approach** that is known as **recursive binary splitting**.
- **Top-down**: it begins at the top of tree and then successively splits the predictor space, each split is indicated via two new branches further down on the tree.
- **Greedy**: at each step of the tree-building process, the best split is made at the particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step.

More Details in Tree Building Process

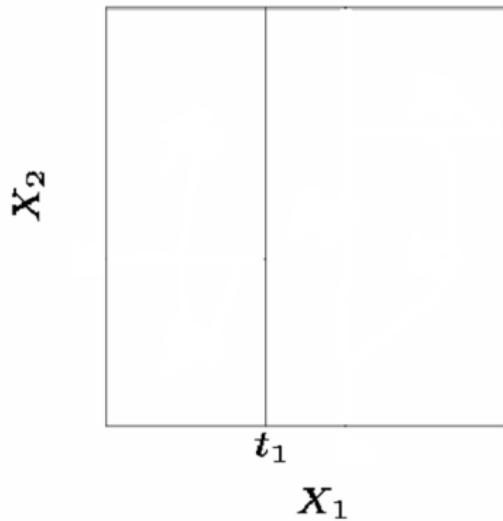
- We first select the predictor X_j and cutpoint s such that splitting the predictor space into regions $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ leads to the **greatest possible reduction in RSS**.
- Then, we repeat the process, searching for the best predictor and cutpoint to further split the data so as to minimize the RSS within each of the resulting regions. Note rather than splitting the entire space, we **split one of the two previously identified regions**. Now we have three regions.
- Again, we look to split one of these three regions further, so as to minimize the RSS. The process continues until a stopping criterion is reached, e.g. we may continue until no region contains no more than five observations.
- Once the regions R_1, \dots, R_J have been created, **how to make predictions?**

Splitting the X Variables



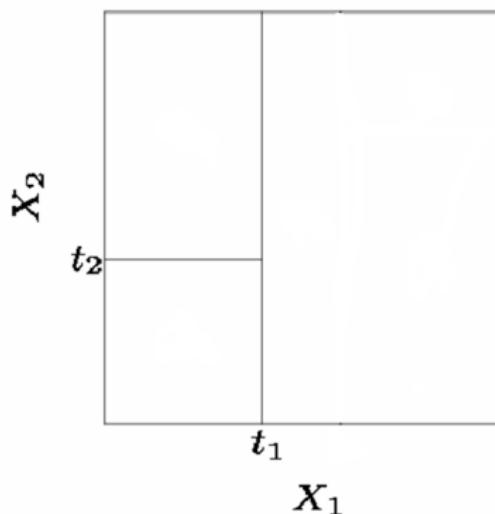
Generally we create the partitions by iteratively splitting one of the X variables into two regions.

Splitting the X Variables



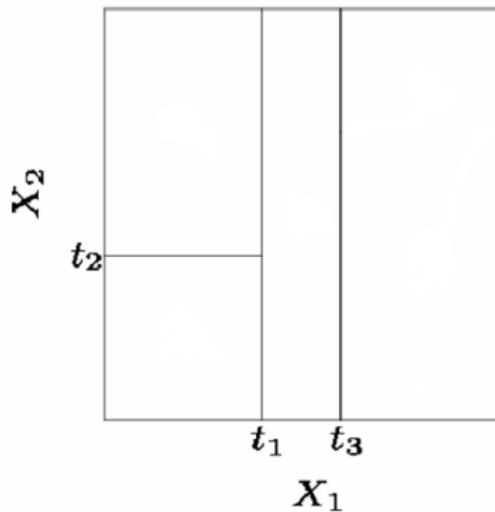
- ① First split on $X_1 = t_1$.

Splitting the X Variables



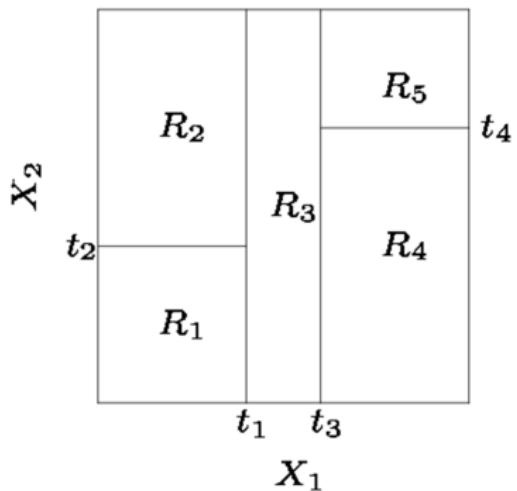
- ① First split on $X_1 = t_1$.
- ② If $X_1 < t_1$, split on $X_2 = t_2$.

Splitting the X Variables



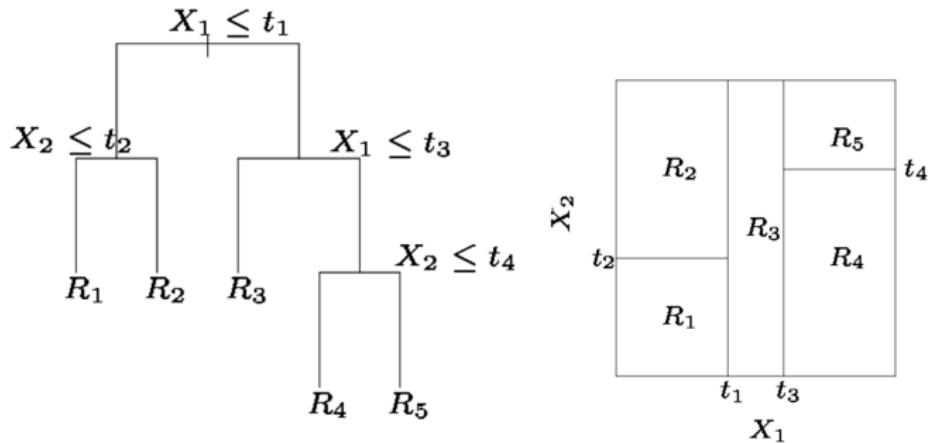
- ① First split on $X_1 = t_1$.
- ② If $X_1 < t_1$, split on $X_2 = t_2$.
- ③ If $X_1 > t_1$, split on $X_1 = t_3$.

Splitting the X Variables



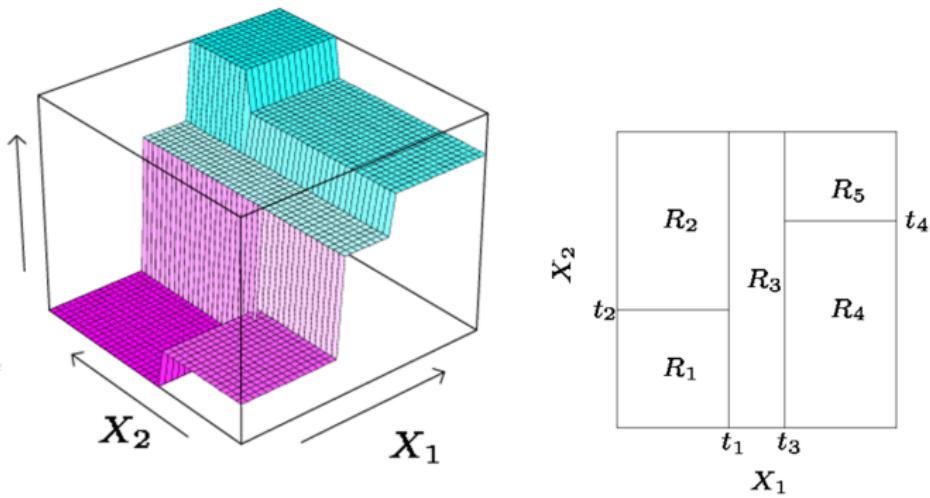
- ① First split on $X_1 = t_1$.
- ② If $X_1 < t_1$, split on $X_2 = t_2$.
- ③ If $X_1 > t_1$, split on $X_1 = t_3$.
- ④ If $X_1 > t_3$, split on $X_2 = t_4$.

A Tree Representation



- When we create partitions this way we can always represent them using a tree structure.
- This provides a very simple way to explain the model to a non-expert, i.e. your boss :).

A Tree Representation



- Between each of the five regions we give a different prediction for Y .
- Within each of the five regions we give the same prediction for Y .

Pruning a Tree

Pruning a Tree

- The tree building process may produce good predictions on the training set, but is likely to **overfit** the data, leading to poor test set performance.
- We can grow a very tree T_0 and then prune it back in order to obtain a subtree.
- Cost complexity pruning, we consider a sequence of trees indexed by a tuning parameter α . For each value of α there corresponds a subtree $T \subset T_0$ such that

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

is as small as possible. $|T|$ indicates the number of terminal nodes of the tree T , R_m is the rectangle corresponding to the m -th terminal node, and \hat{y}_{R_m} is the mean of the training observation in R_m .

Choose the Optimal Subtree

- α is the tuning parameter that controls a trade-off between the subtree's complexity and its fit to the training data.
- Optimal α can be obtained using cross validation.
- We then return to the full dataset and obtain the subtree corresponding to the optimal α .

Algorithm for Building a Regression Tree

- ① Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.
- ② Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtree, as a function of α .
- ③ Use K -fold cross validation to choose α . That is, divide the training observation into K folds. For each $k = 1, \dots, K$:
 - a Repeat Steps 1 and 2 on all but the k -th fold of the training data.
 - b Evaluate the mean squared prediction error on the data in the left-out k -th fold, as a function of α .

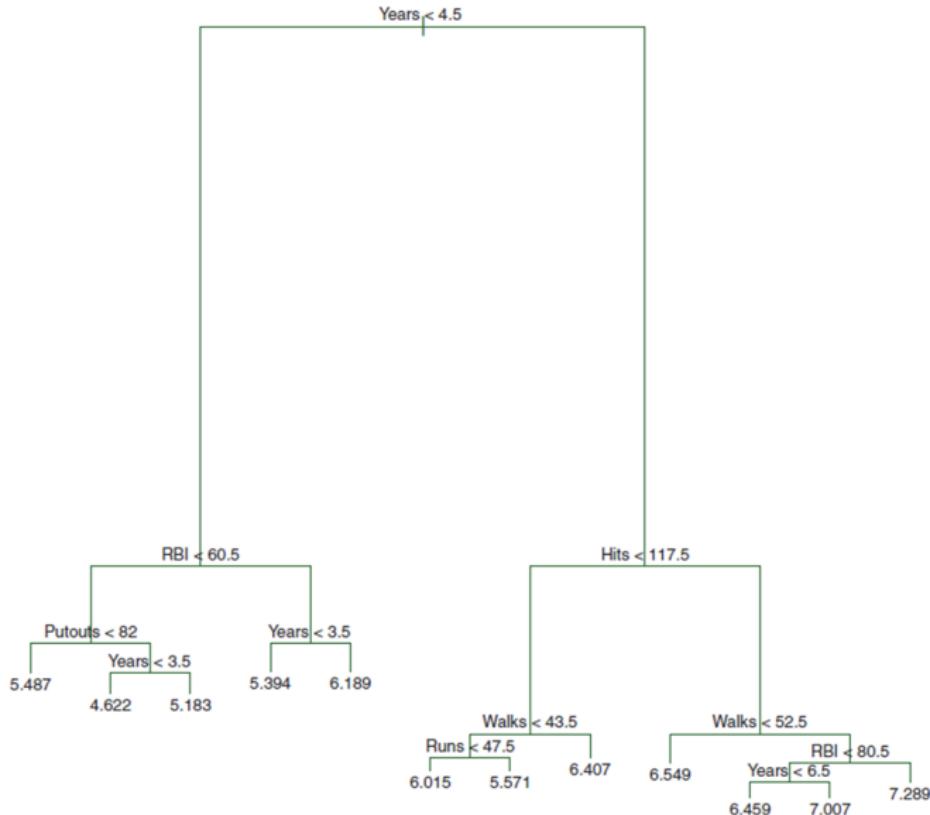
Average the results for each value of α and pick α to minimize the average error.

- ④ Return the subtree from Step 2 that corresponds to the chosen value of α .

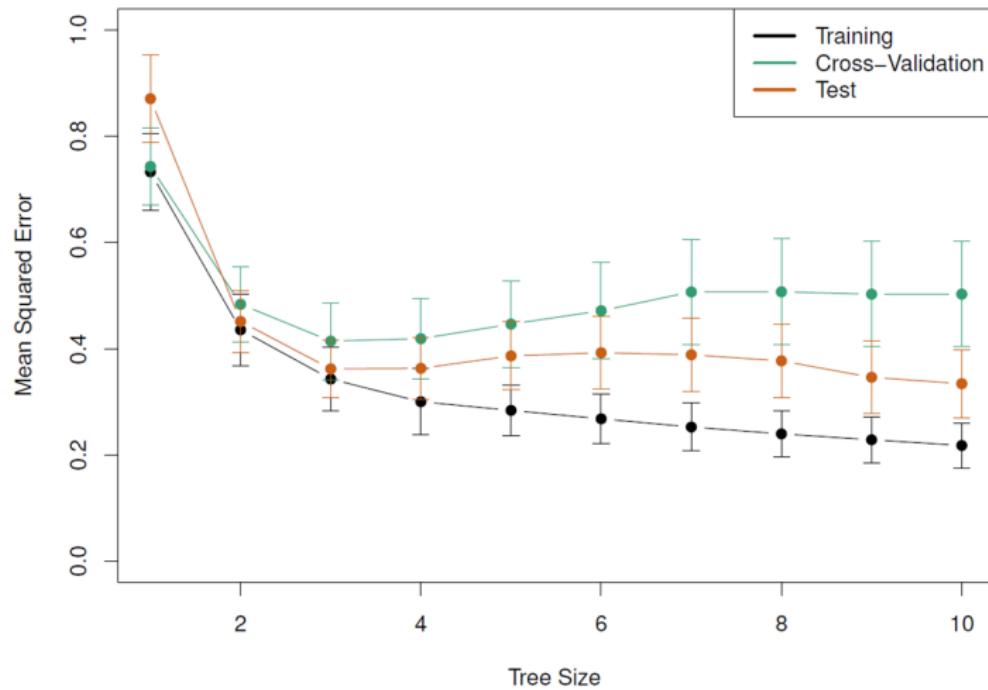
Baseball Data Continued

- Randomly split the data: 132 training observations and 131 observations in the test dataset.
- We then build a large regression tree on the training data and varied α to create subtrees with different number of terminal nodes.
- Finally, a six-fold cross validation is implemented to estimate the cross-validated MSE of the trees as a function of α .

Baseball Data Regression Tree



Cross Validation (See Page 5)



Classification Tree

Classification Tree

- Qualitative response rather than quantitative response.
- For a classification tree, we predict that each observation belongs to the **most common occurring class** of training observations in the region to which it belongs.
- We use **recursive binary splitting** to grow a classification tree.
- Recall that in a regression tree, we define the cost complexity criterion:

$$C_\alpha(|T|) = \sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha|T| = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|,$$

where $Q_m(T) = \frac{1}{N_m} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2$ is the MSE of the regression in the m -th terminal node (region) and $N_m = \#\{x_i \in R_m\}$ is the number of observations within the m -th region.

Classification Tree

- “ Q_m ” is not suitable in the classification tree, we need to substitute “ Q_m ” by some alternative measures:
- A natural alternative RSS is the **misclassification error**, which is the fraction of training observations in the region that do not belong to the most common class:

$$Q_m(T) = \frac{1}{N_m} \sum_{i:x_i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)},$$

where $k(m) = \arg \max_k \hat{p}_{mk}$ and $\hat{p}_{mk} = \frac{1}{N_m} \sum_{i:x_i \in R_m} I(y_i = k)$ represents the proportion of training observations in the m -th region that are from the k -th class.

Two Other Measures

- However classification error is not sufficiently sensitive for tree growing and in practice two other measures are preferable.
- **Gini index:**

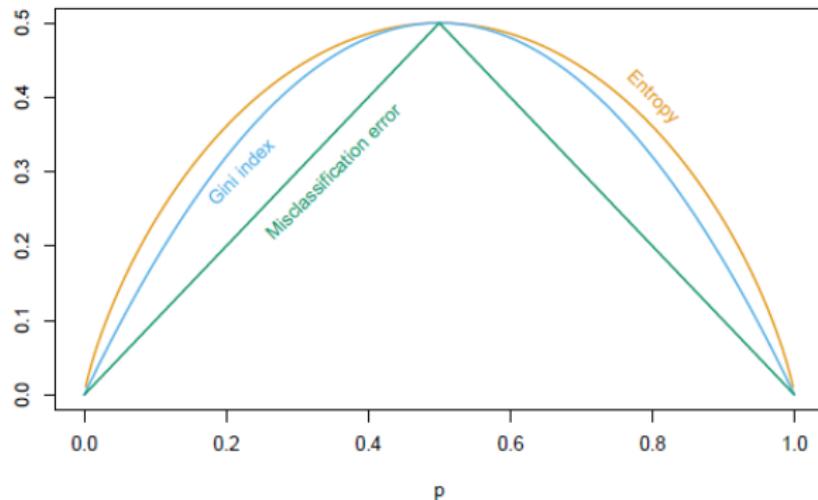
$$Q_m(T) = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

- a measure of total variance across the K classes.
- Gini index is referred to as a measure of **node purity**- a small value indicates a node contains predominately observations from a single class.
- **Cross-entropy (deviation):**

$$Q_m(T) = - \sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}.$$

- These two measures are used to evaluate the **quality of a particular split**, since they are more sensitive to node purity than the classification error rate.

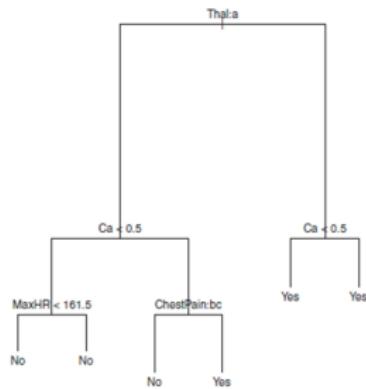
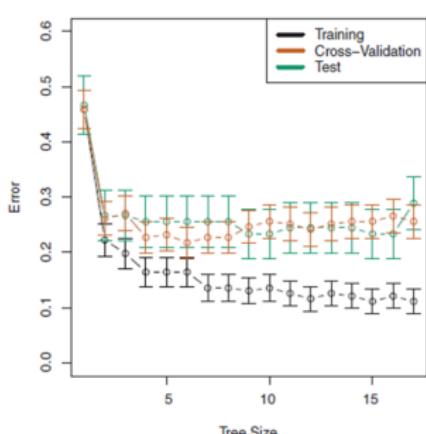
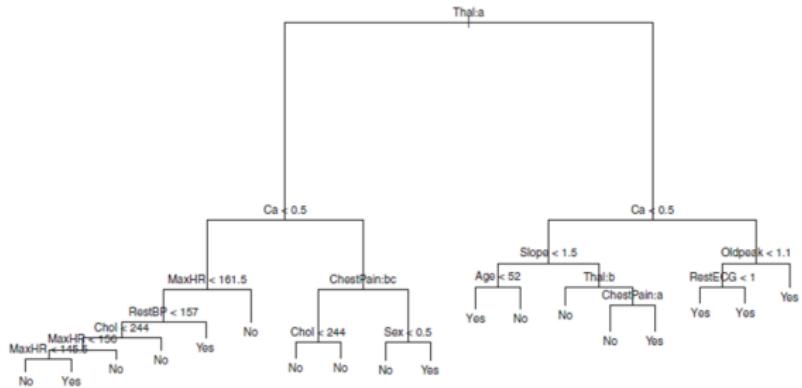
Comparison Among Three Measures



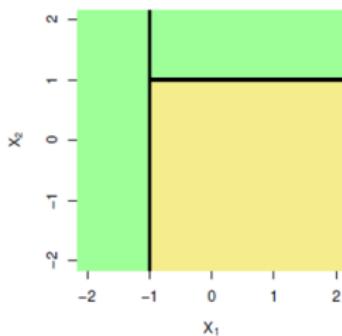
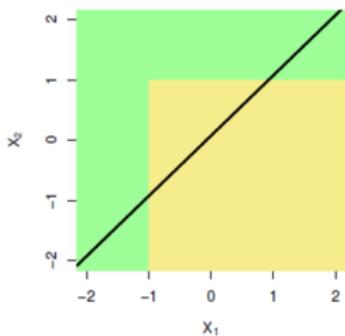
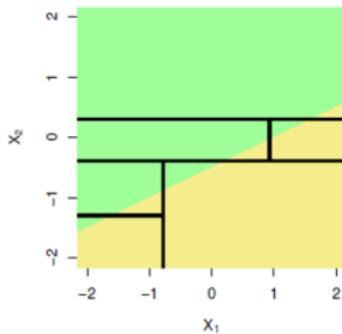
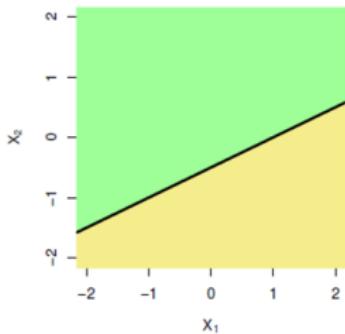
- For $K = 2$, the three measures are $1 - \max(p, 1 - p)$, $2p(1 - p)$ and $-p \log(p) - (1 - p) \log(1 - p)$, respectively.
- The cross-entropy and the Gini index are differentiable and hence more amendable to numerical optimization.

Heart Data

- Binary outcome HD for 303 patients who presented with chest pain.
- An outcome value of Yes indicates the presences of heard disease based on an angiographic test, while No means no heart decease.
- There are 13 predictors including Age, Sex, Chol (a cholesterol measurement) and other heard and lung function measurements.
- CV yields a tree with six terminal nodes, see next page.



Tree vs Linear Models



Top row: true linear boundary; Bottom row: true nonlinear boundary.

Advantages and Disadvantages of Trees

- Trees are easy to explain to people. They are even easier to explain than linear regression.
- Some people believe that decision trees more closely mirror human decision making than do the regression and classification seen in previous lectures.
- Trees can be displayed graphically and easily interpreted even by a non-expert.
- Trees can easily handle qualitative predictors without the need to create dummy variables.
- Trees do not have the same level of predictive accuracy as some of the other regression and classification approaches seen in previous lectures.

Bagging

A Variance Reduction Approach

- The bootstrap is an extremely powerful idea but what does it have to do with statistical learning?
- Suppose that we have a procedure (such as trees, neural nets etc.) that has high variance.
- An ideal way to reduce variance would be to take many samples from the population, build a separate prediction model using each sample and average the resulting predictions i.e.

$$\hat{f}_{\text{average}}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^b(x).$$

Bagging

- Of course, as discussed previously, we cannot take multiple different samples from the population.
- However, we can use the bootstrap approach which does the next best thing by taking repeated samples from the training data.
- We therefore end up with B different training data sets.
- We can train our method on each data set and then average all the predictions i.e.

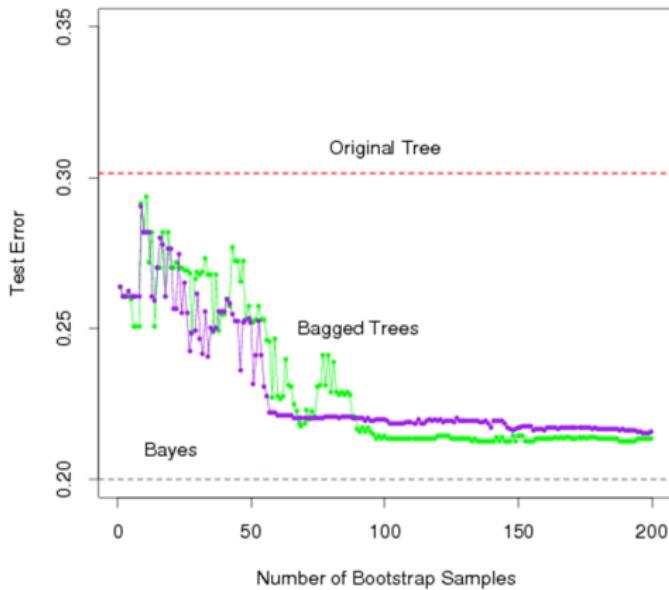
$$\hat{f}_{bag}(x) = \frac{1}{B} \hat{f}^{*b}(x).$$

- This approach is called **Bagging** or **Bootstrap Aggregation**.

Bagging Regression and Classification Tree

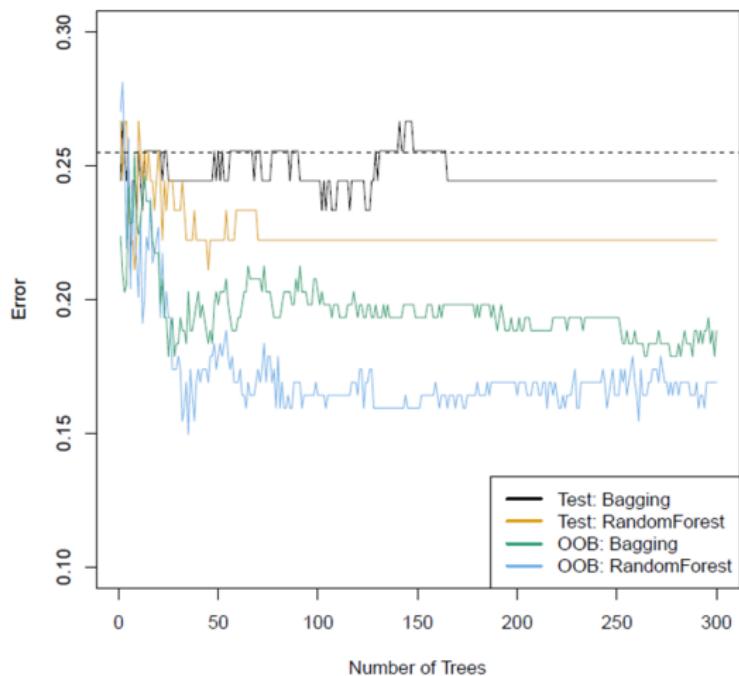
- The previous consider the regression trees.
- **Classification tree**: for each test observation, we record the class predicted by each of B trees and take a **majority vote**: the overall prediction is the **most commonly occurring class** among B predictions. (We use majority vote for HD).
- Or you can **take the average of B probabilities** and assign to the class with the **highest averaged probability**.

Simulation to Compare Error Rates



- Here the green line represents a simple majority vote approach.
- The purple line corresponds to averaging the probabilities.
- Both methods do far better than a single tree and get close to the Bayes error rate.

Bagging the Heart Data



Out-of-Bag Error Estimation

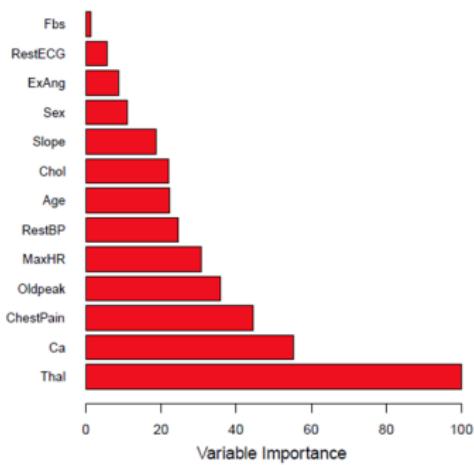
- Recall that the key to bagging is that trees are repeatedly fit to bootstrapped subsets of the observations. One can show that on average, each bagged tree makes use of around $2/3$ of the observations. **Why?**
- The remaining $1/3$ of the observations not used to fit a given bagged tree are referred to as *Out-of-Bag* (OOB) observations.
- We can predict the response for the i -th observation using each of the trees in which that observation was OOB. This may yield around $B/3$ predictions for the i -th observation, which we average.
- It can be shown that OOB error is essentially the LOOCV error for bagging if B is large. (Free way to do LOOCV!).

Random Forest

- Random forest provides an improvement over bagged trees by way of a small tweak that **decorrelates** the trees. This **reduce the variance** when we average the tree.
- As in bagging, we build a number of decision trees on bootstrapped training samples.
- But when building these decision trees, each time a split in a tree is considered, **a random selection of m predictors** is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors.
- We choose $m = \sqrt{p}$.

Variable Importance Measure

- Bagging typically results in improved accuracy over prediction using a single tree. However, it can be difficult to interpret the resulting model.
- We record the total amount that the RSS is decreased due to splits over a given predictor, average over all B trees. A large value indicated an important predictor.
- Variable importance plot.



Boosting

Boosting

- Boosting can be applied to many statistical learning approaches. We concentrate on decision trees.
- Recall that bagging involves creating copies of the original training dataset using the bootstrap, and then combining all of the trees to create a single predictive model.
- Each tree is independent of trees based on a bootstrap dataset.
- Boosting works in a similar way, but the trees are grown **sequentially**, each tree is grown using information from previously grown trees.

Boosting Algorithm for Regression Trees

- ① Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
- ② For $b = 1, \dots, B$, repeat:
 - a Fit a tree \hat{f}^b with d splits ($d + 1$ terminal nodes) to the training data (X, r) .
 - b Update \hat{f} by adding in a shrunken version of the new tree:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$

- c Update the residuals,

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$$

- ③ Output the boosted model,

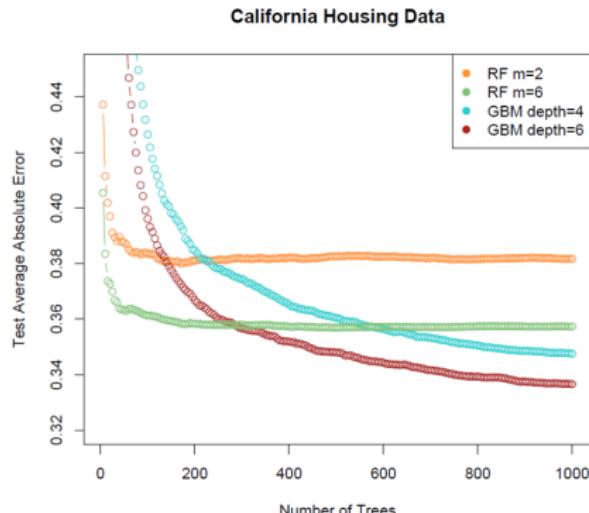
$$\hat{f}(x) = \sum_{b=1}^B \lambda \hat{f}^b(x).$$

Ideas behind

- Unlike fitting a single large decision tree to the data, which amounts to fitting the data hard, the boosting approach instead learns slowly.
- Given the current model, we fit a decision tree to the residuals from the model. We then add this new decision tree into then fitted function in order to update the residuals.
- Each of these trees can be rather small, with just a few terminal nodes, determined by the parameter d in the algorithm.
- By fitting small trees to the residuals, we slowly improve \hat{f} in areas where it does not perform well. This shrinkage parameter λ slows the process down even further, allowing more and different shaped trees to attack the residuals.

Regression Example

- Number of trees B (CV for selecting B); shrinkage parameter λ (typical values are 0.01 or 0.001) and number of splits d (interaction depth) are tuning parameters for boosting.
- See further details in Chapter 10 of ESL.
- R package: **gbm**.
- Gradient boosted models.



Summary

- Decision trees are simple and interpretable. But they can not lead to high prediction accuracy.
- Bagging, random forests and boosting are developed to improve the prediction accuracy of trees.
- Random forests and boosting are among the state-of-art methods for regression or classification (supervised learning). However, their results can be difficult to interpret.
- Further details of random forests and boosting can be learnt from ESL.

ST 443: Machine Learning and Data Mining

Dr. Xinghao Qiao

Columbia House, Room 5.15

x.qiao@lse.ac.uk

Department of Statistics



Office Hours: Tuesday 4:30–5:30pm

Lecture 7

Q1. Maximal Margin Classifier (MMC)

Q2. Support Vector Classifier (SVC)

Q3. Support Vector Machines (SVM)

Support Vector Machines

- Vladimir Vapnik (Applied Mathematician from Soviet Union) with Corinna Cortes at Bell Lab around 1990.
- No probability model, SVM was developed in the computer science community. SVM is considered one of the best “out of box” classifiers.
- Idea: we try and find a plane that **separates** the classes in the features space.
- No such plane? Two solutions:
 - ① *Soften the “separates”.*
 - ② *We enlarge the feature space to make “separates” possible!*

Maximal Margin Classifier

What is a Hyperplane?

- In a p -dimensional space, a **hyperplane** is a flat affine subspace of dimension $p - 1$.
- In general p -dimensional setting, a hyperplane is defined by the equation

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0.$$

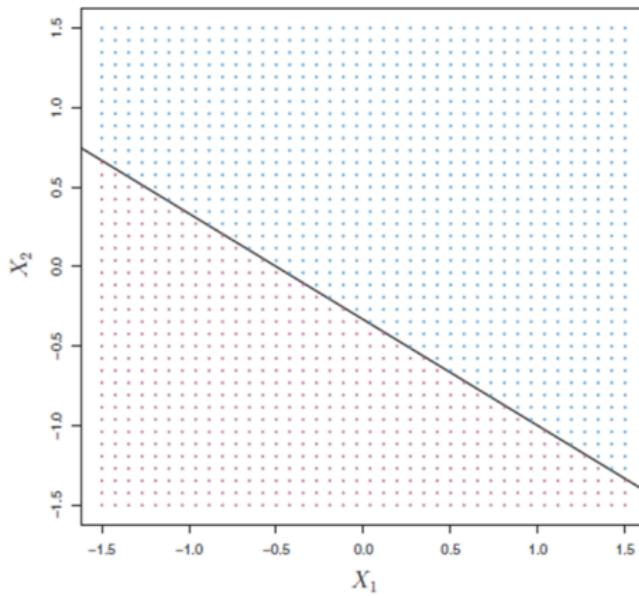
- If $p = 2$, the hyperplane is a line.
- If $\beta_0 = 0$, the hyperplane goes through the origin.
- The vector $\beta = (\beta_1, \dots, \beta_p)^T$ is called the **normal vector**, in a direction orthogonal to the surface of a hyperplane.
- X lies to one side of the hyperplane if

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p > 0,$$

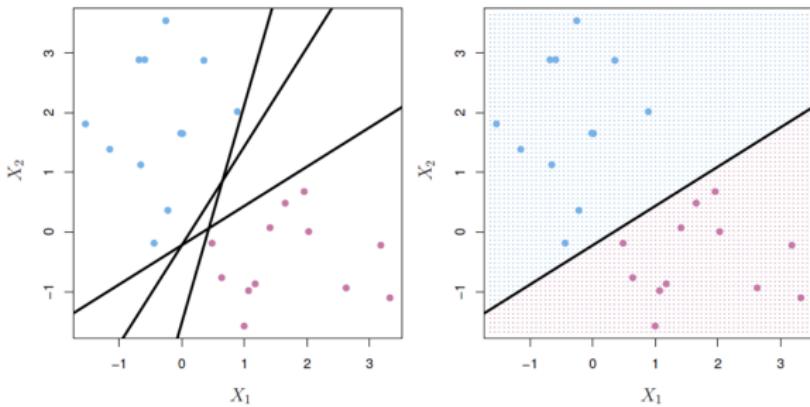
or the other side if the above “ $>$ ” is replaced by “ $<$ ”.

Hyperplane 2-Dimensions

- The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown. The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$ and the purple region $1 + 2X_1 + 3X_2 < 0$. (See the whiteboard).

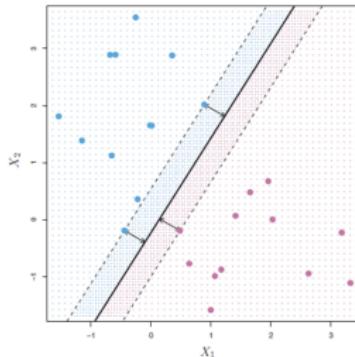


Separating Hyperplanes



- Let $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$, then $f(X) > 0$ for points on one side of the hyperplane and $f(X) < 0$ for points on the other.
- Let $Y_i = +1$ for blue and $Y_i = -1$ for purple, then $Y_i f(X_i) > 0$ for all $i = 1, \dots, N$. Then $f(X) = 0$ defines a **separating hyperplane!**
- Given a test observation x_0 , we can use the sign of $f(x_0)$ to assign the class.
- Magnitude of $f(x_0)$. Large values? Small values?

Maximal Margin Hyperplane



- Compute the perpendicular distance from each training observation to a given separating hyperplane; the smallest such distance is the **margin**.
- Among all separating hyperplanes, find the one makes the **largest margin** between two classes.
- Consider the following optimization problem:

$$\underset{\beta_0, \beta_1, \dots, \beta_p}{\text{maximize}} M$$

subject to $\sum_{j=1}^p \beta_j^2 = 1, y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M \text{ for all } i = 1, \dots, n.$

Maximal Margin Classifier

- We can then classify a test observation based on which side of the maximal margin hyperplane it lies. This is known as the **Maximal Margin Classifier**.
- Given a test observation \mathbf{x}_0 we assign the class based on the sign of

$$f(\mathbf{x}_0) = \beta_0 + \beta_1 x_{01} + \cdots + \beta_p x_{0p}.$$

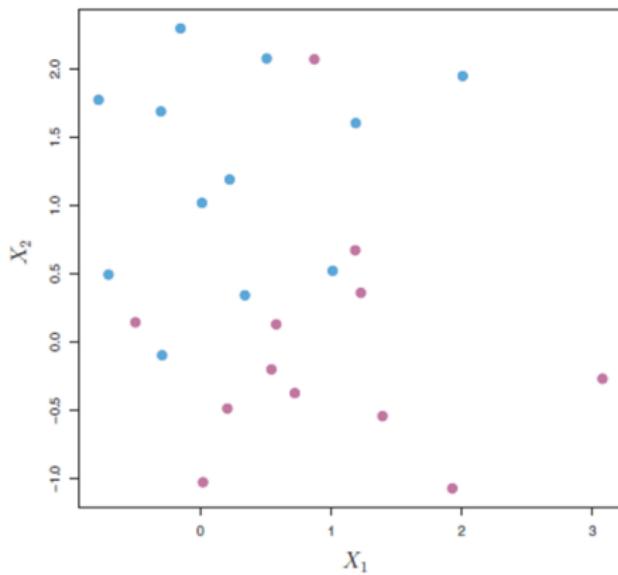
- Actually $\sum_{j=1}^p \beta_j^2 = 1$ is not really a constraint on the hyperplane. One can show that this **constraints the perpendicular distance** from the i -th observation to the hyperplane given by

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}).$$

- Thus the constraints ensure that each observation is on the correct sign of the hyperplane and at least a distance M (width of margin) from the hyperplane.
- The optimization is a **convex quadratic** problem and can be solved efficiently. The function **`svm()`** in `e1071` package solves the problem.

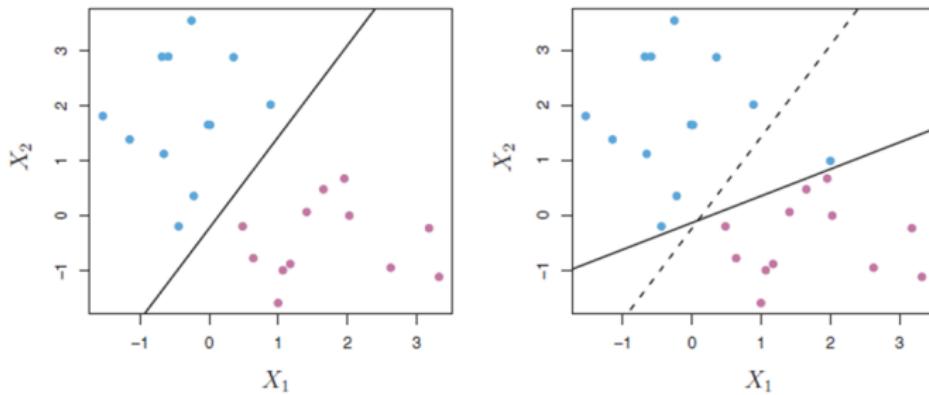
The Non-Separable Case

- The two classes are not separable by a hyperplane and so the maximal marginal classifier cannot be used.
- We develop a hyperplane that **almost separates** the classes, using a so-called **soft margin**.



Support Vector Classifier

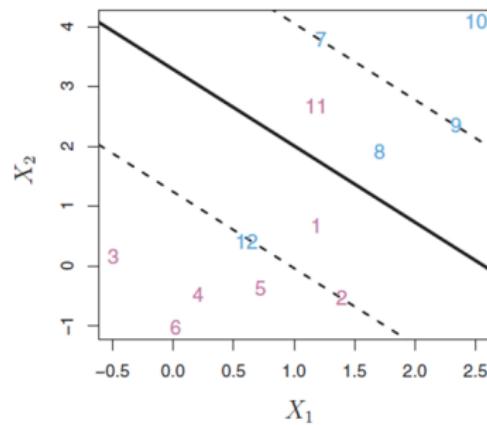
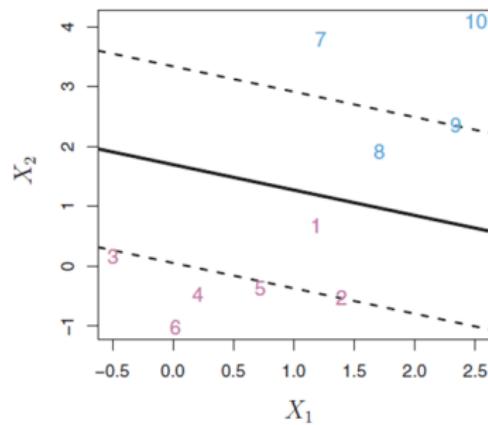
Noisy Data



- It is possible that the data are separable, but noisy. This leads to a dramatic change in the maximal margin hyperplane.
- The maximal hyperplane is **extremely sensitive** to a change in a single observation. **What does this suggest?**
- A classifier based on a hyperplane **does not perfectly separate** the two classes?
 - ➊ Greater **robustness** to individual observations.
 - ➋ Better classification of **most** of the training observations.

Support Vector Classifier (SVC)

- Support vector classifier is also called a **soft margin** classifier.
- We allow some observations to be on the **incorrect side of the margin**, or even the **incorrect side of the hyperplane**.
- Observations on the wrong side of the hyperplane correspond to training observations that are misclassified by support vector classifier.



More on SVC (1)

- SVC separates most of the training observations into the two classes, but many misclassify a few observations. Consider the optimization problem

$$\underset{\beta_0, \beta_1, \dots, \beta_p, \epsilon_1, \dots, \epsilon_n}{\text{maximize}} M$$

$$\text{subject to } \sum_{j=1}^p \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \sum_{i=1}^n \epsilon_i \leq C,$$

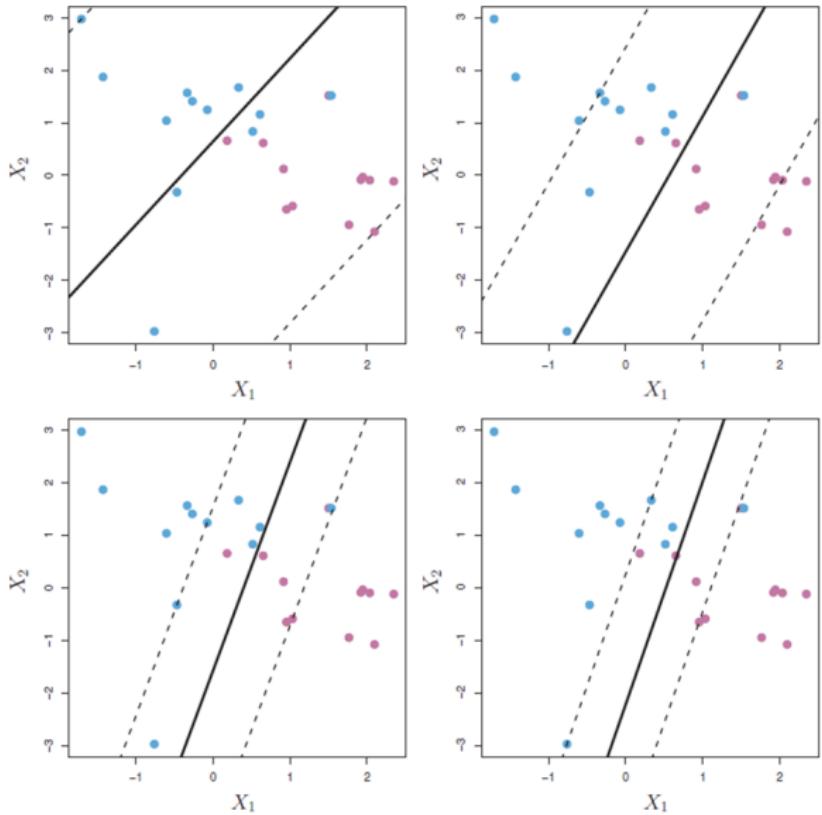
where C is a nonnegative tuning parameter (**budget**). M is the width of the margin.

- Here $\epsilon_1, \dots, \epsilon_n$ are **slack variables** that allow individual observations to be on the wrong side of the margin or the hyperplane.

More on SVC (2)

- ϵ_i tells us where the i -th observation is located, relative to the hyperplane and relative to the margin.
 - ① $\epsilon_i = 0$, the i -th observation is on the **correct sides of the margin**.
 - ② $\epsilon_i > 0$, then i -th observation is on the **wrong side of the margin**, and if $\epsilon_i > 1$ then it is on the **wrong sides of the hyperplane**.
- C determines the number and severity of the violations to the margin.
Budget for the amount that the margin can be violated by n observations.
 - ① $C = 0$, it must be $\epsilon_1 = \dots = \epsilon_n = 0$.
 - ② For $C > 0$, no more than C observations can be on the wrong side of the hyperplane.
 - ③ As C increases, we become more tolerant of violations to the margin, so the margin will widen.
 - ④ C controls the variance and bias tradeoff. **How?**

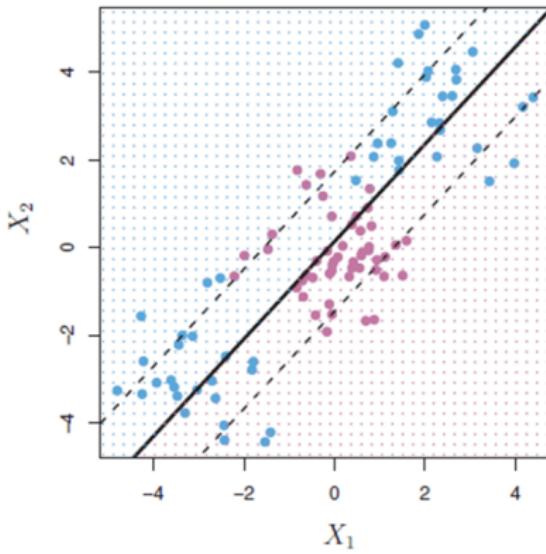
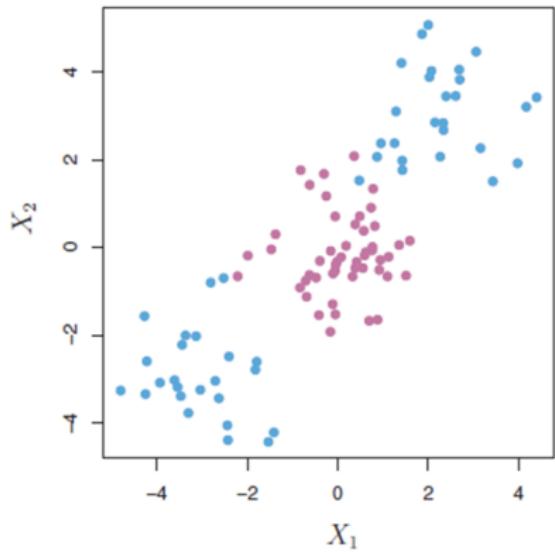
C is a Regularization Parameter



More on SVC (3)

- From the optimization, it turns out that only observations that either lie on the margin or violate the margin affect the hyperplane.
- Observations that lie directly on the margin, or on the wrong sides of the margin for their class are known as **support vectors**. (See the figure).
- SVC decision rule is based only on a potentially small subset of the training observations, which means that it is quite robust to the behaviour of observations that are far away from the hyperplane.
- This is very different from LDA, but logistic regression also has low sensitivity to observations far from the decision boundary.

Linear Boundary can Fail!



Support Vector Machines

Nonlinear Decision Boundary

- Enlarge the space of features by including

$$X_1, X_1^2, X_2, X_2^2, \dots, X_p, X_p^2.$$

- Then the SVM optimization problem becomes

$$\underset{\beta_0, \beta_{11}, \beta_{12}, \dots, \beta_{p1}, \beta_{p2}, \epsilon_1, \dots, \epsilon_n}{\text{maximize}} M$$

$$\text{subject to } \sum_{j=1}^p \sum_{k=1}^2 \beta_{jk}^2 = 1,$$

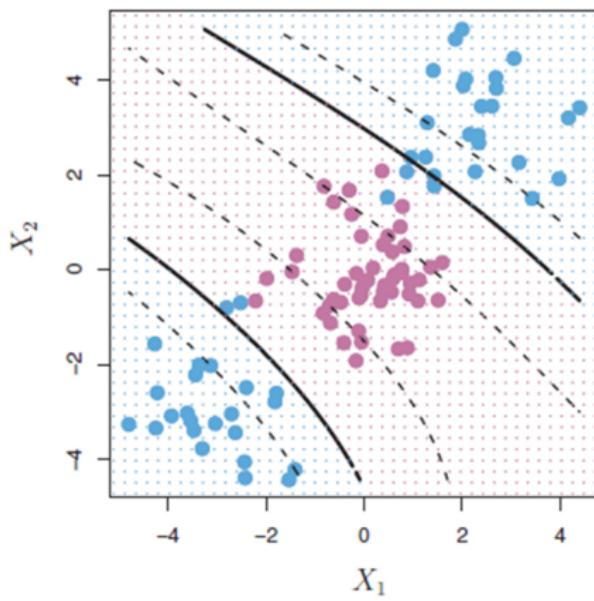
$$y_i \left(\beta_0 + \sum_{j=1}^p \beta_{j1} x_{ij} + \sum_{j=1}^p \beta_{j2} x_{ij}^2 \right) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \sum_{i=1}^n \epsilon_i \leq C,$$

- Nonlinear decision boundaries.
- Many possible methods to enlarge the feature space.

Cubic Polynomials

- Here we use a basis expansion of cubic polynomials.
- From 2 to 9 variables.
- The SVC in the enlarged space solves the problem in the lower-dimensional space.



Nonlinearities and Kernels

- Polynomials can lead to computations unmanageable (especially in high dimensions).
- The main idea is to enlarge our feature space to accommodate a non-linear boundary between classes. The **kernel approach** is simply an efficient computational approach for enacting this idea.
- **Inner products** in support vector classifier.
- Consider $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T \in \mathbb{R}^p, i = 1, \dots, n$.

Inner Products and Support Vectors

- Inner product between vectors

$$\langle \mathbf{x}_i, \mathbf{x}_{i'} \rangle = \sum_{j=1}^p x_{ij} x_{i'j}.$$

- The linear SVC can be represented as

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^n \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle,$$

where the parameters $\alpha_1, \dots, \alpha_n$ and β_0 can be estimated using $n(n - 1)/2$ inner products $\langle \mathbf{x}_i, \mathbf{x}_{i'} \rangle$ between all pairs of training observations.

- It turns out that α_i is nonzero only for support vectors in the solution. Let \mathcal{S} be the collection of indices of these support points, then

$$f(\mathbf{x}) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle.$$

Any sort of sparsity?

Kernels

- We can replace the inner product with its generalization in the form of

$$K(\mathbf{x}_i, \mathbf{x}_{i'}),$$

where K is some function that we will refer to as a **kernel**.

- Non linear mapping from input space to **high dimensional feature space**:
 $\Phi : \mathbf{X} \rightarrow F$.
- Define $K : \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{R}$, called **kernel**, such that

$$\Phi(\mathbf{x})^T \cdot \Phi(\mathbf{x}') = K(\mathbf{x}, \mathbf{x}'),$$

where K is often interpreted as a similarity measure.

- **Flexibility**: K can be chosen arbitrarily so long as the existence of Φ is guaranteed (**positive definite symmetric** (PDS) condition).

Example: Polynomial Kernels

- Polynomial kernel of degree d

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \left(c + \sum_{j=1}^p x_{ij} x_{i'j} \right)^d,$$

computes the inner-products needed for d -dimensional polynomials, i.e.

$\binom{p+d}{d}$ basis functions. Try it for $p = 2$ and $d = 2$.

- Why is K PDS for $p = 2$ and $d = 2$?

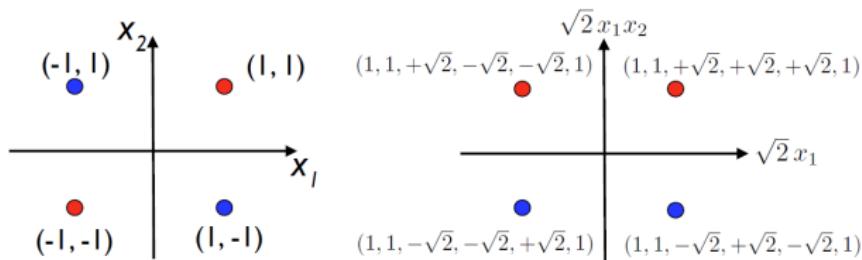
Example: Polynomial Kernels

- Polynomial kernel of degree d

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \left(c + \sum_{j=1}^p x_{ij} x_{i'j} \right)^d,$$

computes the inner-products needed for d -dimensional polynomials, i.e. $\binom{p+d}{d}$ basis functions. Try it for $p = 2$ and $d = 2$.

- Why is K PDS for $p = 2$ and $d = 2$?
- Consider second-degree polynomial kernel with $c = 1$.



Linearly non-separable

Linearly separable by

$$x_1 x_2 = 0.$$

Kernels and SVM

- A **kernel** is a function that quantifies the similarity of two observations.
- SVC considers the linear kernel with

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^p x_{ij} x_{i'j}.$$

- Non-linear polynomial kernel of degree d

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \left(1 + \sum_{j=1}^p x_{ij} x_{i'j} \right)^d.$$

- SVC combined with a non-linear kernel leads to the resulting classifier known as **Support Vector Machines (SVM)** in the form of

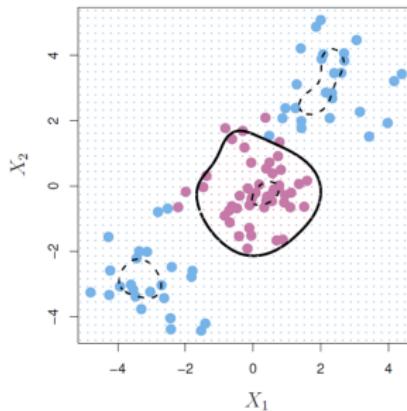
$$f(\mathbf{x}) = \beta_0 + \sum_{i \in S} \alpha_i K(\mathbf{x}, \mathbf{x}_i).$$

SVM with Radial Kernel

- Consider

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \exp\left(-\gamma \sum_{j=1}^p (x_{ij} - x_{i'j})^2\right).$$

- How does radial kernel work? **Local behavior**, in the sense only nearby training observations have an effect on the class label of test observations.



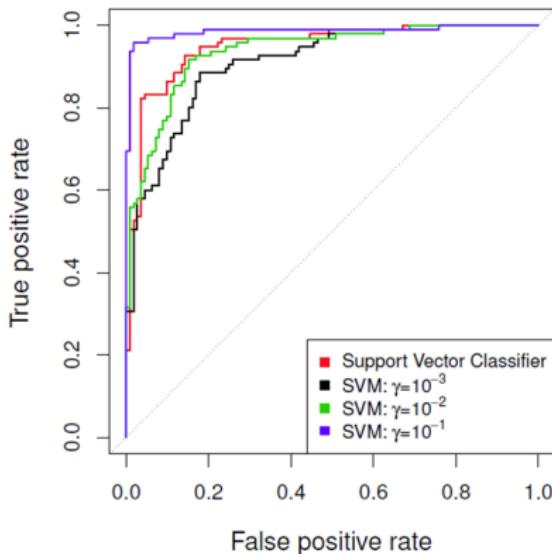
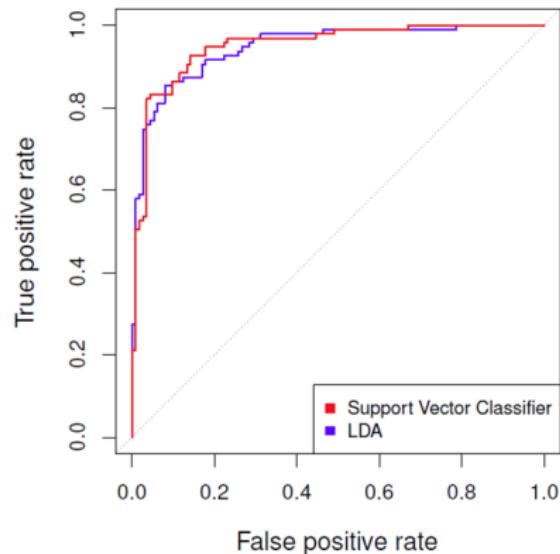
How can one deal with very high dimensional feature space? Controls variance by squashing down most dimensions severely.

Advantages of Kernel Approaches

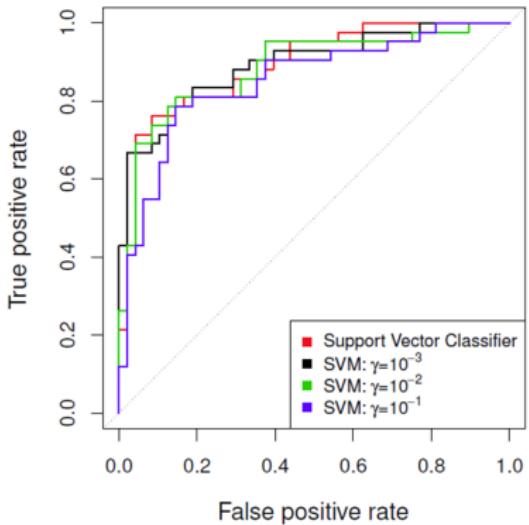
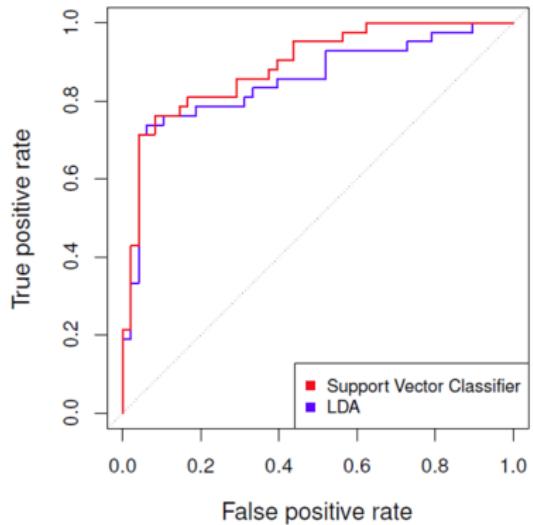
- Less computational cost.
- It amounts to the fact that using kernels, one only need compute $K(\mathbf{x}_i, \mathbf{x}_{i'})$ for all $\binom{n}{2}$ distinct pairs i and i' .
- This is done without explicitly working on the enlarged features space, especially in the high dimensional $n < p$ setting (the computation is intractable!)

Example: Heart Data

ROC curve is obtained by changing the threshold 0 to threshold t in $f(X) > t$. Below is the ROC curves on training data.



Heart Test Data



SVM on More than Two Classes

- **One versus All.** Fit K different 2-class SVM classifiers

$$\hat{f}_k(x), k = 1, \dots, K.$$

Each class versus the rest. Classify x_0 to the class for which $\hat{f}_k(x_0)$ is the largest.

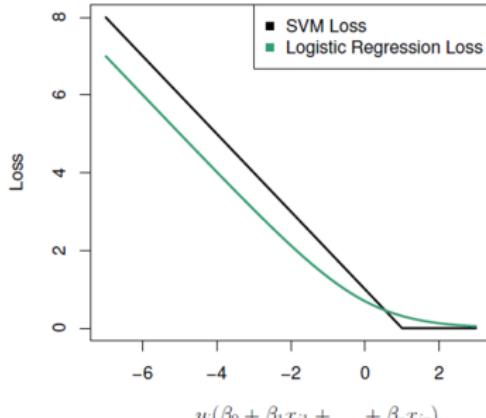
- **One versus One.** Fit all $\binom{K}{2}$ pairwise classifier $\hat{f}_{kl}(x)$. Classify x_0 to the class that is most frequently assigned among all pairwise classifications.

SVM vs Logistic Regression

- Given $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$, SVC can be rewritten as

$$\underset{\beta_0, \beta_1, \dots, \beta_p}{\text{minimize}} \sum_{i=1}^n \max \{0, 1 - y_i f(x_i)\} + \lambda \sum_{j=1}^p \beta_j^2.$$

- Loss (hinge loss) + ℓ_2 penalty.
- The loss is very similar to the loss in logistic regression (negative log likelihood).
- See Session 9.5 of ISL and Chapter 12 of ELS for further details.



SVM vs Logistic Regression (LR)

- When classes are nearly separable, SVM does better than LR.
- When not, LR with ridge penalty and SVM are similar.
- For probability? LR!
- For nonlinear boundaries? Kernel SVM!

ST 443: Machine Learning and Data Mining

Dr. Xinghao Qiao

Columbia House, Room 5.15

x.qiao@lse.ac.uk

Department of Statistics



Office Hours: Tuesday 4:30–5:30pm

Lecture 8

Q1. Principal component analysis

Q2. K-means clustering

Q3. Hierarchical clustering

Unsupervised Learning

- **Supervised learning:** regression and classification. We typically have access to a set of features X_1, \dots, X_p , measured on n observations and a response Y . The goal is to predict Y using X_1, \dots, X_p .
- Here we focus on **unsupervised learning**, where we observe only the features, X_1, \dots, X_p . We are interested not in prediction, because do not have an associated response variable Y .
- The goal of unsupervised learning: is there an informative way to visualize the data? Can we discover subgroups among the variable or among the observations?

Unsupervised Learning

- **Supervised learning:** regression and classification. We typically have access to a set of features X_1, \dots, X_p , measured on n observations and a response Y . The goal is to predict Y using X_1, \dots, X_p .
- Here we focus on **unsupervised learning**, where we observe only the features, X_1, \dots, X_p . We are interested not in prediction, because do not have an associated response variable Y .
- The goal of unsupervised learning: is there an informative way to visualize the data? Can we discover subgroups among the variable or among the observations?
- Unsupervised learning is more subjective than supervised learning.
- It is more often to obtain unlabelled data from lab instrument or computer than labelled data, which can require human intervention.

Principal component analysis (PCA)

PCA

- PCA provides a low-dimensional representation of a dataset. It finds the sequence of linear combinations of the variables that have **maximal variance** and are **mutually uncorrelated**.
- Data visualization tool.

PCA Details

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

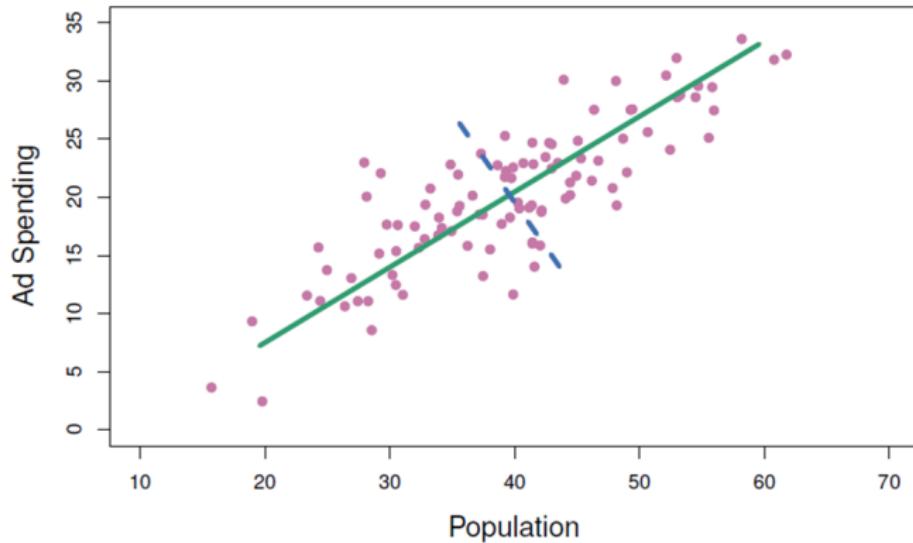
$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$$

that has the largest variance under the constraint $\sum_{j=1}^p \phi_{j1}^2 = 1$.

- We call $\phi_{11}, \dots, \phi_{p1}$ as the **loadings of the first principal component**.
The principal component loading vector

$$\boldsymbol{\phi}_1 = (\phi_{11}, \phi_{21}, \dots, \phi_{p1})^T.$$

PCA Example



100 different cities, green solid line indicated the first PC direction and the blue dashed line indicates the second PC direction.

Estimation for 1st PC

- We have the n by p observed data matrix \mathbf{X} . Each of the variables has been centred to have zero mean.
- First PC loading vector solves the optimization problem

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

- We refer to Z_1 as the first **principal component**, with values z_{11}, \dots, z_{n1} defined as

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{ip}.$$

Further Principal Components

- The second PC is the linear combination of X_1, \dots, X_p that has maximal variance among all linear combinations that are uncorrelated with Z_1 .
- The second PC score $z_{12}, z_{22}, \dots, z_{n2}$ takes the form

$$z_{i2} = \phi_{12}x_{i1} + \phi_{22}x_{i2} + \cdots + \phi_{p2}x_{ip},$$

where ϕ_2 is the 2nd PC loading vector with elements $\phi_{12}, \phi_{22}, \dots, \phi_{p2}$.

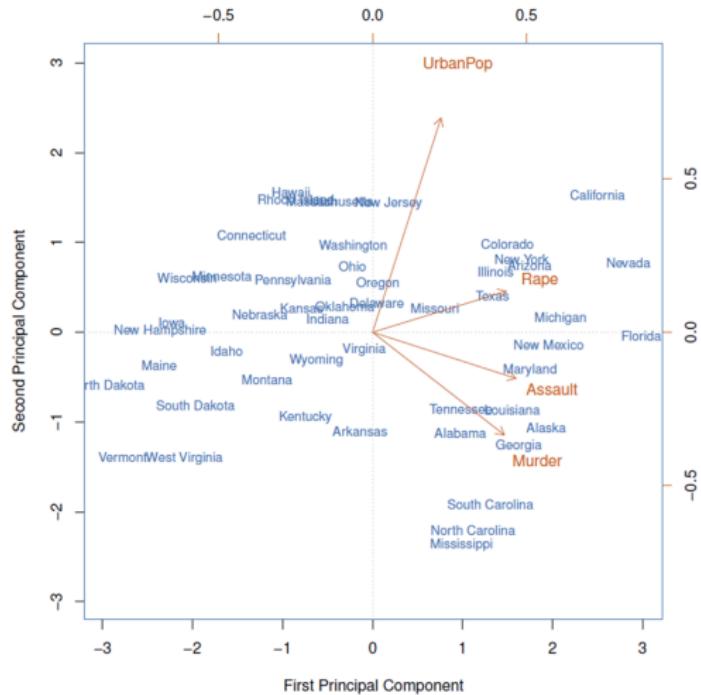
- It turns out that constraining Z_2 uncorrelated with Z_1 is equivalent to having the 2nd PC direction $\phi_2 = (\phi_{12}, \phi_{22}, \dots, \phi_{p2})^T$ orthogonal to 1st PC direction $\phi_1 = (\phi_{11}, \phi_{21}, \dots, \phi_{p1})^T$.
- And so on.

USA Arrest Data

- For each of the fifty states in the USA, the dataset contains the number of arrests per 100,000 residents for each of three crimes: **Assault**, **Murder**, **Rape**. Also we record **UrbanPop** (the percent of the population in each state living in urban areas).
- $n = 50, p = 4$.
- PCA was performed after standardization each variable to have mean zero and standard deviation one.

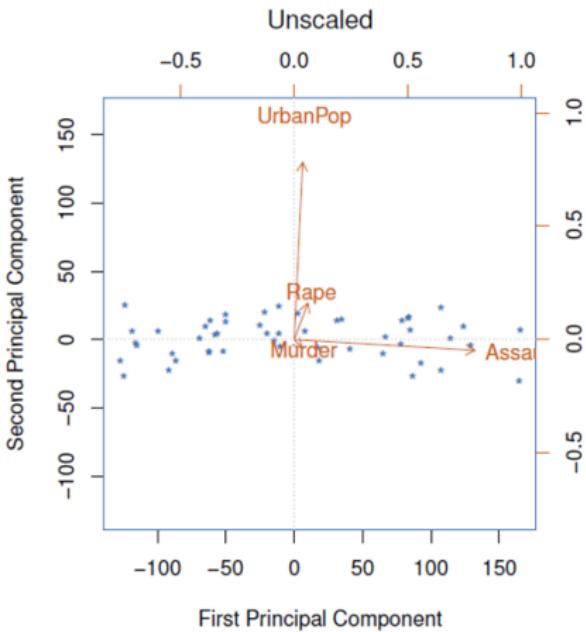
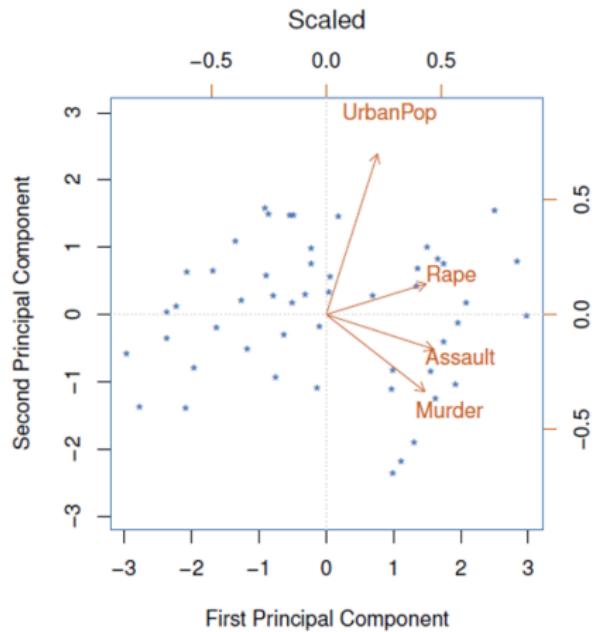
PCA Plot: Biplot with 2 PC and PC Loadings

The blue states names represent the scores for the first two PCs; the orange indicate the first two PC loadings (The loading for Rape on the first PCS is 0.54 and its loading the second PC is 0.17, the Rape is centered at the point (0.54, 0.17)).



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

Scaling is Important

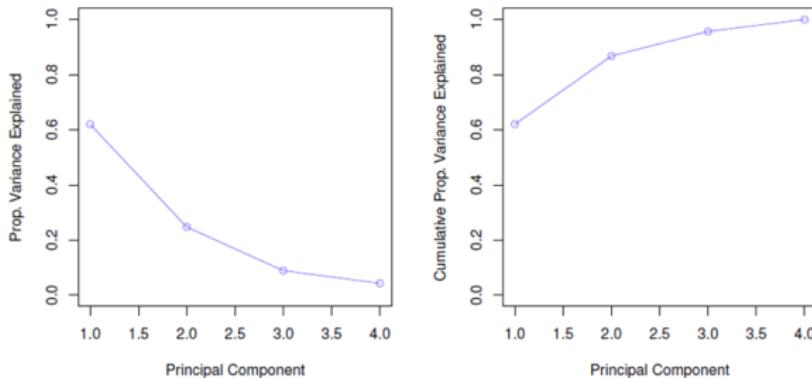


Proportion Variance Explained (PVE)

- The PVE of the m -th PC is given by

$$\frac{\sum_{i=1}^n z_{im}^2}{\sum_{j=1}^p \sum_{i=1}^n x_{ij}^2}$$

- The PVEs sum to one. Follow the textbook for details.



- How many PC are sufficient as a summary of our data?

K-means clustering

Clustering

- **Clustering** refers to a very broad set of techniques for finding **subgroups** or **clusters**, in a dataset.
- We seek a partition of n observations of the data into distinct groups so that the observations within each group are quite similar to each other.
- To make this concrete, we need define what it means for two or more observations to be similar or different.
- Indeed this is often a domain specific consideration that must be made based on knowledge of the data being studied.
- PCA vs clustering method
 - PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the **variance**.
 - Clustering looks to find **homogeneous** subgroups among the observations.

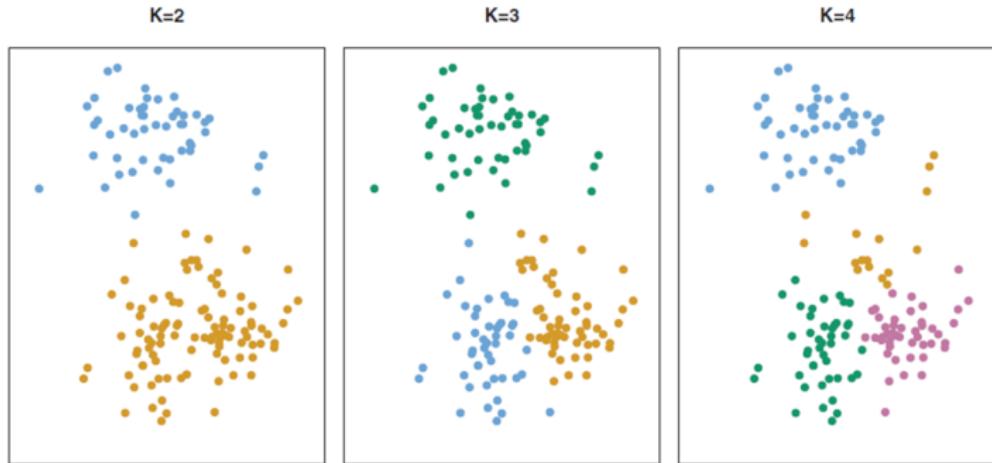
Clustering for Market Segmentation

- Application of clustering arises in marketing.
- We may have access to a large number of measurements (e.g. median household income, occupation, distance from nearest urban area, and so forth) for a large number of people.
- Our goal is to perform **market segmentation** by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a product.
- The task of performing market segmentation amounts to clustering the people in the dataset.

Two Clustering Methods

- **K-means clustering:** we aim to partition the observation into a pre-specified number of clusters.
- **Hierarchical clustering:** the number of clusters are not known in advance; we end up with a tree-like representation of the observations, called a **dendrogram**, which allows us to view at once the clusterings obtained from each possible number of clusters from 1 to n .

K-means Clustering



A simulation dataset with 150 observations with $p = 2$. Panels show that the results of applying K -means clustering with different values of K , the number of clusters.

Details of K-means Clustering

- Let C_1, \dots, C_K denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:
 - $C_1 \cup C_2 \cup \dots \cup C_K = \{1, \dots, n\}$. In other words, each observation belongs to at least one of the K clusters.
 - $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observations belongs to more than one cluster.
- For instance, if the i -th observation is in the k -th cluster, than $i \in C_k$.

Details of K-means Clustering (Continued)

- Idea to perform K-means is that a **good** clustering is one for which the **within-cluster variation** is as small as possible.
- The within-cluster variation for C_k is a measure $W(C_k)$ of the amount by which the observations within a cluster differ from each other.
- Hence we wish to solve the problem

$$\min_{C_1, \dots, C_K} \left\{ \sum_{k=1}^K W(C_k) \right\}.$$

- In words, this formula says that we want to partition the observations into K clusters such that the total within-cluster variation, summed over all K clusters, is as small as possible.

Within-cluster Variation

- There are many possible ways to define this concept, but by far the most common choice involves **squared Euclidean distance**.

$$W(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2,$$

where $|C_k|$ denotes the number of observations in the k -th cluster.

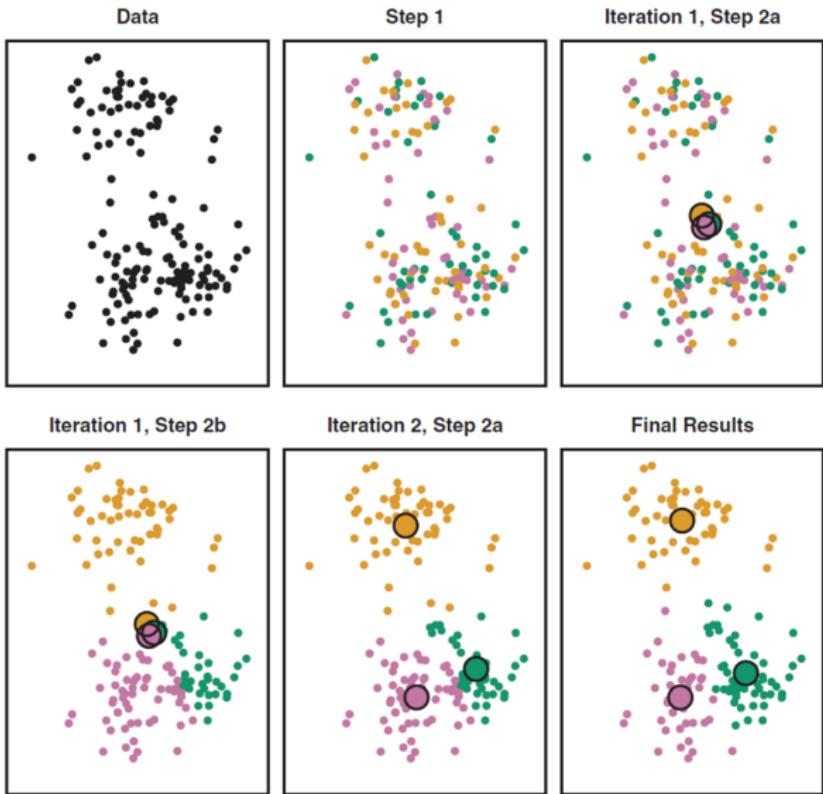
- Finally, we aim to solve the following optimization problem for K -means

$$\min_{C_1, \dots, C_K} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}.$$

K-means Clustering Algorithm

- ① Randomly assign a number, from 1 to K , to each of the observations. These serve as initial cluster assignments for the observations.
- ② Iterate until the cluster assignments stop changing:
 - a For each of the K clusters, compute the cluster *centroid*. The k -th cluster centroid is the vector of the p features means for the observations in the k -th cluster.
 - b Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

Example



More on the Algorithm

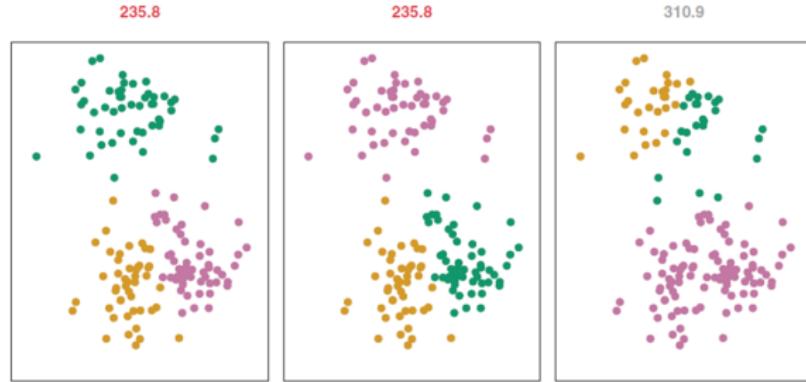
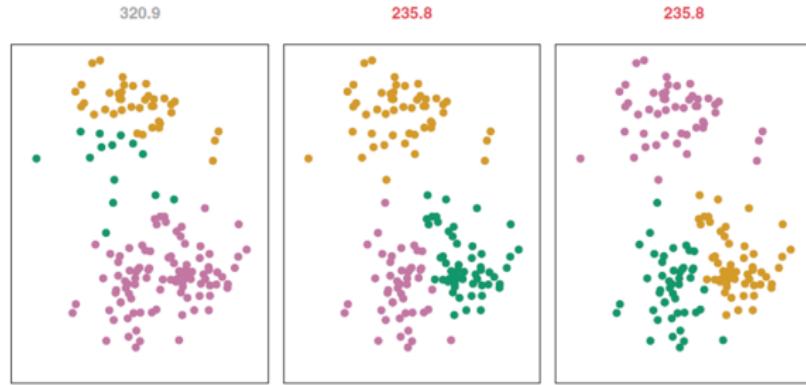
- This algorithm is guaranteed to decrease the value of the objective at each step. **Why?**

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j'})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^p (x_{ij} - \bar{x}_{kj})^2,$$

where $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$ is the mean for feature j in Cluster C_k .

- However, it is not guaranteed to give the global minimum.

Example: Different Starting Values



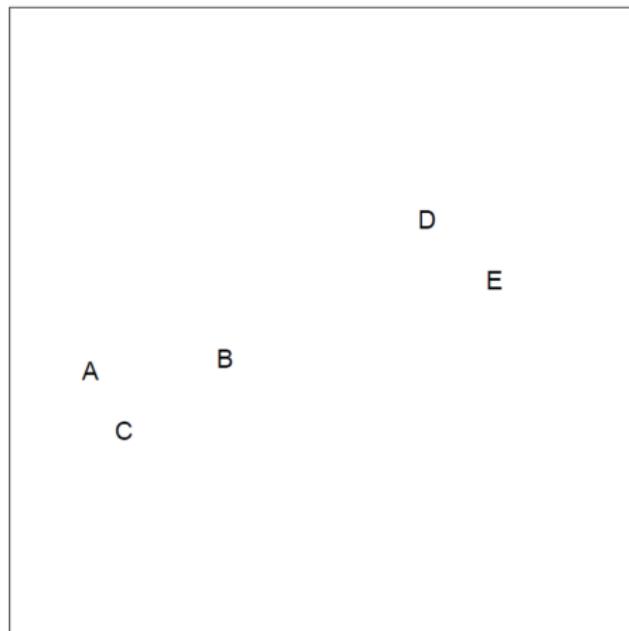
Hierarchical clustering

Hierarchical Clustering

- **K -means clustering** requires us to pre-specify the number of clusters K .
- **Hierarchical clustering** is an alternative approach which does not require that we commit to a particular choice of K . Another advantage is hierarchical clustering results in an attractive tree-based representation of the observations called a **dendrogram**.
- Here, we describe *bottom up* or *agglomerative* clustering. This is the most common type of hierarchical clustering and refers to the fact that a **dendrogram** is built starting from the leaves and combining clusters up to the trunk.

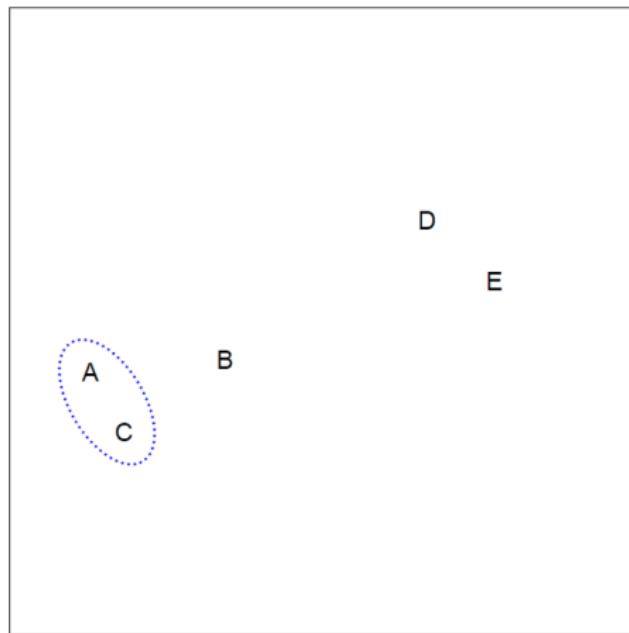
The Idea of Hierarchical Clustering

We constructs a hierarchy in a “bottom-up” way:



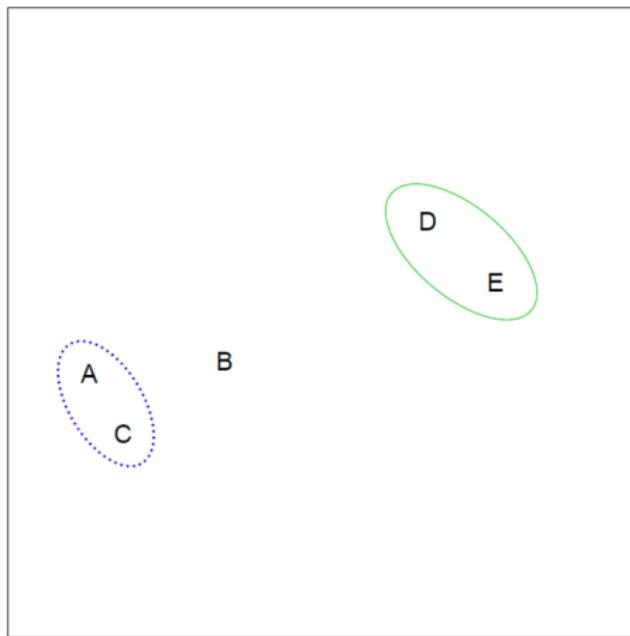
The Idea of Hierarchical Clustering

We constructs a hierarchy in a “bottom-up” way:



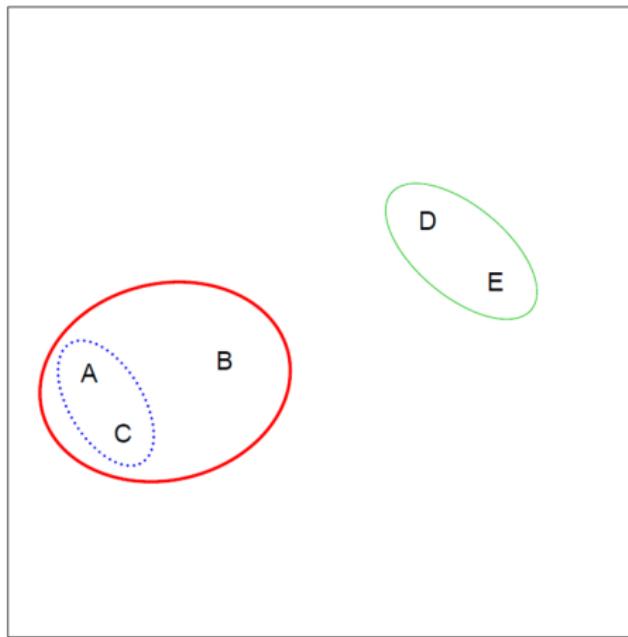
The Idea of Hierarchical Clustering

We constructs a hierarchy in a “bottom-up” way:



The Idea of Hierarchical Clustering

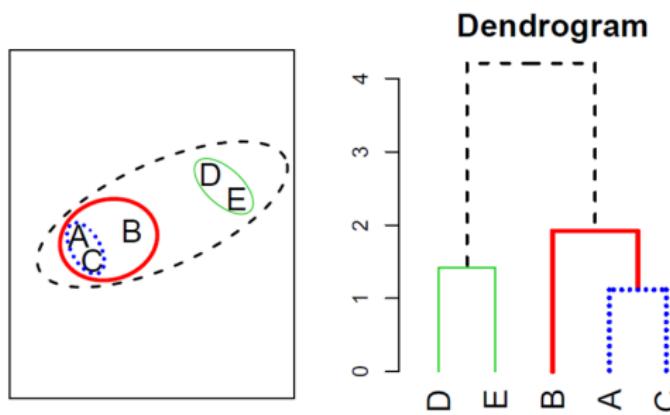
We constructs a hierarchy in a “bottom-up” way:



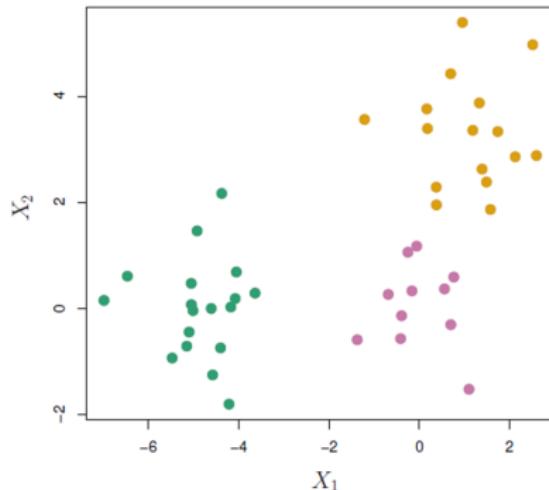
Hierarchical Clustering Algorithm

The algorithm can be summarized as

- ① Start with each point in its own cluster.
- ② Identify the closest two clusters and merge them.
- ③ Repeat Step 2.
- ④ Stop when all points are in a single cluster.

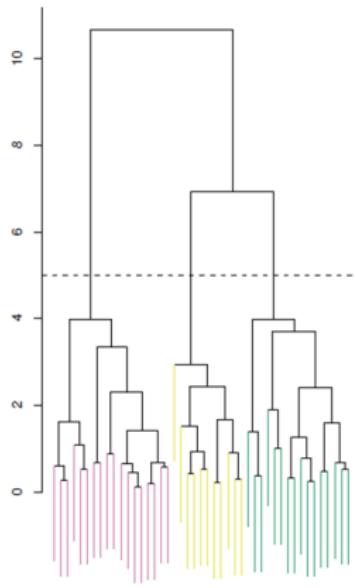
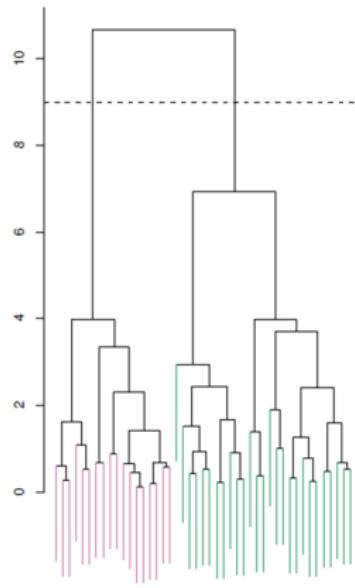
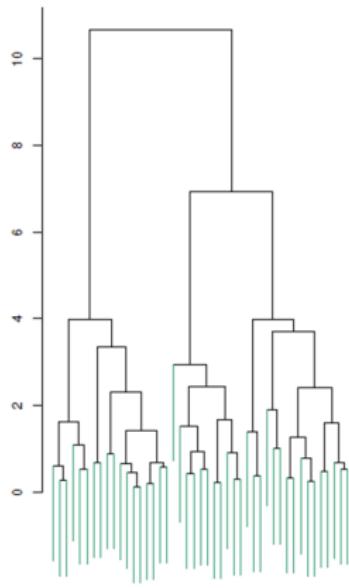


An Simulation Example



45 observations in 2-dimensional feature space. In reality, there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown.

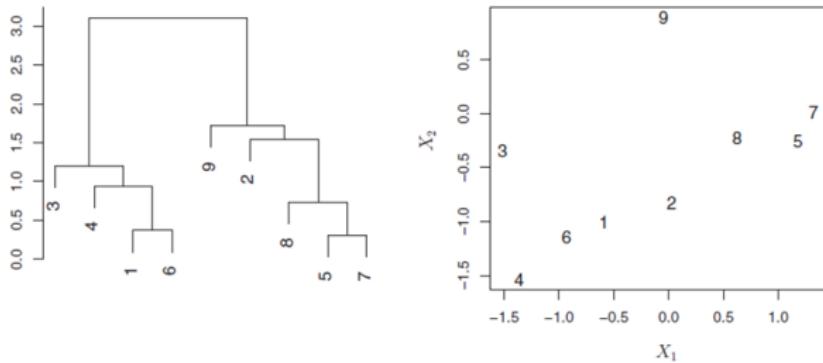
Application of Hierarchical Clustering



More about the Figure

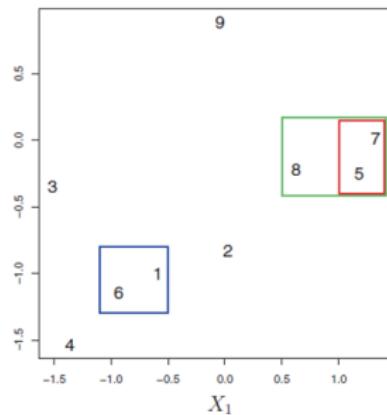
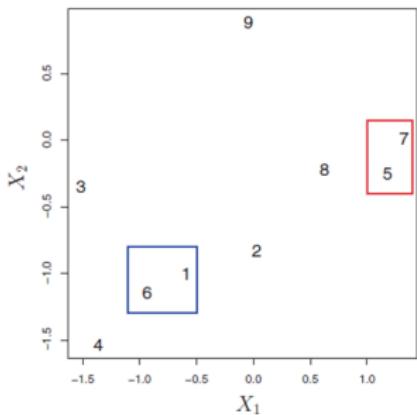
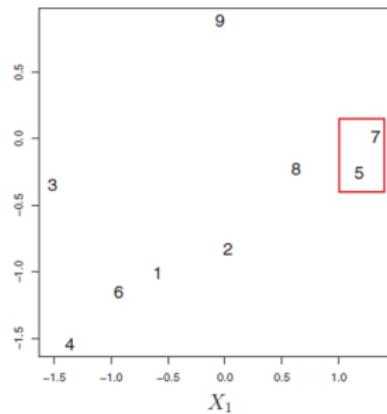
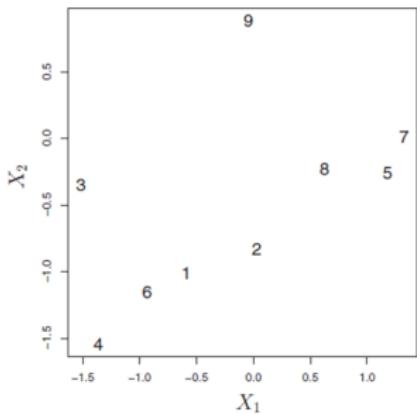
- Each leaf of the dendrogram represents one of the 45 observations.
- The height of fusing (on vertical axis) indicates how different the two observations are.
- The height of the cut to the dendrogram serves the same role as K in K -means clustering: it controls the number of clusters obtained.
- One single dendrogram can be used to obtain any number of clusters.
- In practice, people often look at the dendrogram and select by eye a sensible number of clusters, based on the heights of the fusion and the number of clusters desired.

More about the Figure



- An illustration of how to properly interprets a dendrogram with nine observations in two dimensional space.
- Observations 5 and 7 are quite similar to each other, as are observations 1 and 6.
- However, observations 9 is **no more similar** to observation 2 than it is to observations 8, 5 and 7, even though observations 9 and 2 are close together in terms of horizontal distance.
- This is because observations 2, 8, 5 and 7 all fuse with observations at the same height around 1.8.
- We **cannot** draw conclusions about the similarity of two observations based on their proximity along the horizontal axis.

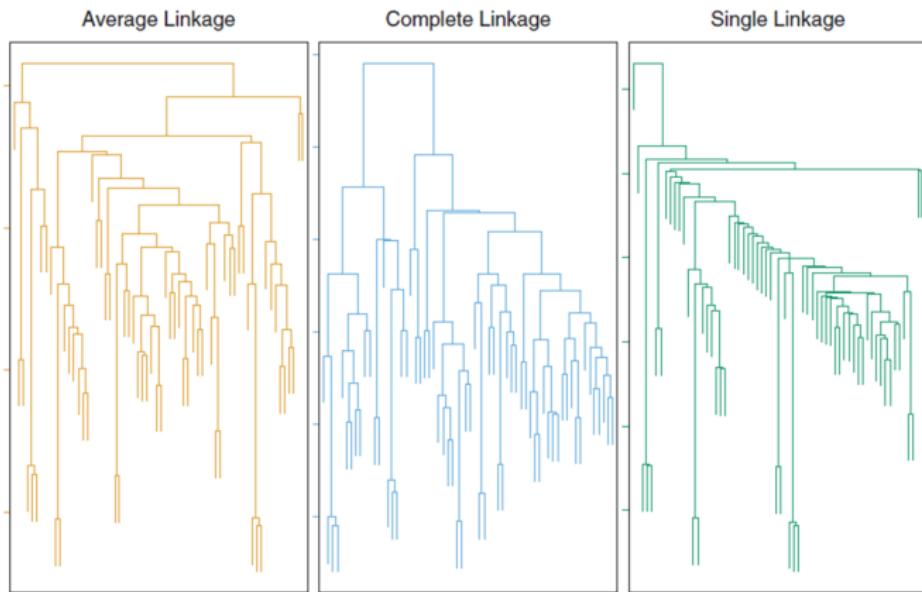
Merges in the Example



Types of Linkage

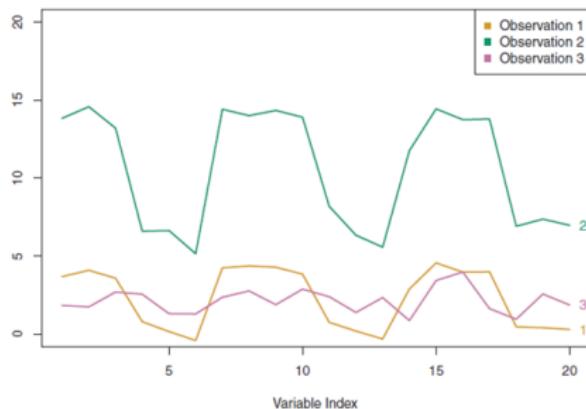
- **Complete Linkage:** Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
- **Single Linkage:** Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observation in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended trailing clusters in which single observations are fused one-at-a-time.
- **Average Linkage:** Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observation in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
- **Centroid Linkage:** Dissimilarity between the centroid for cluster A (a mean vector of length p) and centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Example



Average and complete linkage tend to yield **more balanced clusters**, whereas single linkage tends to yield extended clusters to which single leaves are fused one by one.

Choice of Dissimilarity Measure



- So far, we have used **Euclidean distance**.
- An alternative is **correlation-based distance** which considers two observations to be similar if their features are highly correlated.
- This is an unusual use of correlation, which is normally computed between variables, here it is computed between the observation profiles for each pair of observations.
- Correlation-based distance focuses on the **shapes** of observations profiles rather than the magnitudes.

Marketing Data Example

- Online retailer interested in clustering shoppers based on their past shopping histories. The goal is to identify **subgroups of similar shoppers**.
- It is more reasonable to use correlation-based distance than Euclidean distance.

Practical Issues in Clustering

Small decisions with big consequences.

- **Scaling of the variables matters!** Should the observations or features first be standardized in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of K -means clustering, how many clusters should we look for the data? (See Chapter 13 of ESL for more details).
- In the case of hierarchical clustering.
 - ① What dissimilarity measure should be used?
 - ② What type of linkage should be used?
 - ③ Where should we cut the dendrogram in order to obtain clusters?

Conclusions

- Unsupervised learning is important for understanding the variation and grouping structure of a set of unlabelled data and can be **useful pre-processor for supervised learning!**
- It is more difficult than supervised learning since there is no output and no objective function.
- Hot research topic in Statistics and Computer Sciences, some advanced topics include **self-organizing maps, independent component analysis, spectral clustering**. See Chapter 14 of ELS for more details.