Introduction to Unsupervised Learning

Gabriel Maher
gdmaher@stanford.edu

Alexander Ioannidis
ioannidis@stanford.edu

Institute for Computational and Mathematical Engineering,
Stanford University

ICME

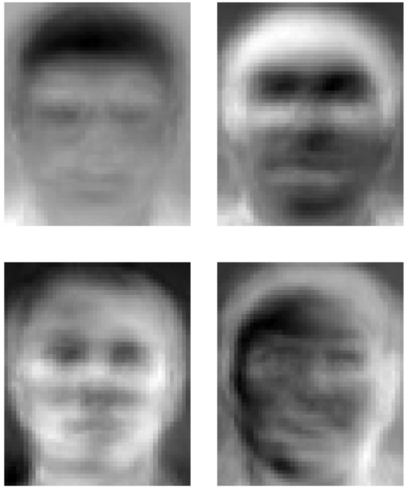
Adapted from slides by
K. Bergen and A. Ioannidis

Unsupervised Learning

- *Unsupervised learning*: a set of statistical tools for data for which only features/inputs are available
  - We have X's but no associated labels Y
  - Goal: discover interesting patterns/properties of the data
    - e.g. for visualizing or interpreting high-dimensional data

Unsupervised Learning

- Sample applications:
  - Given a collection of text documents, identify sets of documents about the same topic
  - Given high-dimensional facial images, find a compact representation as inputs for a facial recognition classifier



(AT&T Laboratories Cambridge)

Unsupervised Learning

- Why is unsupervised learning challenging?
  - Exploratory data analysis – goal is not as clearly defined
  - Difficult to assess performance – “right answer” unknown
  - Working with high-dimensional data

ICME

Unsupervised Learning

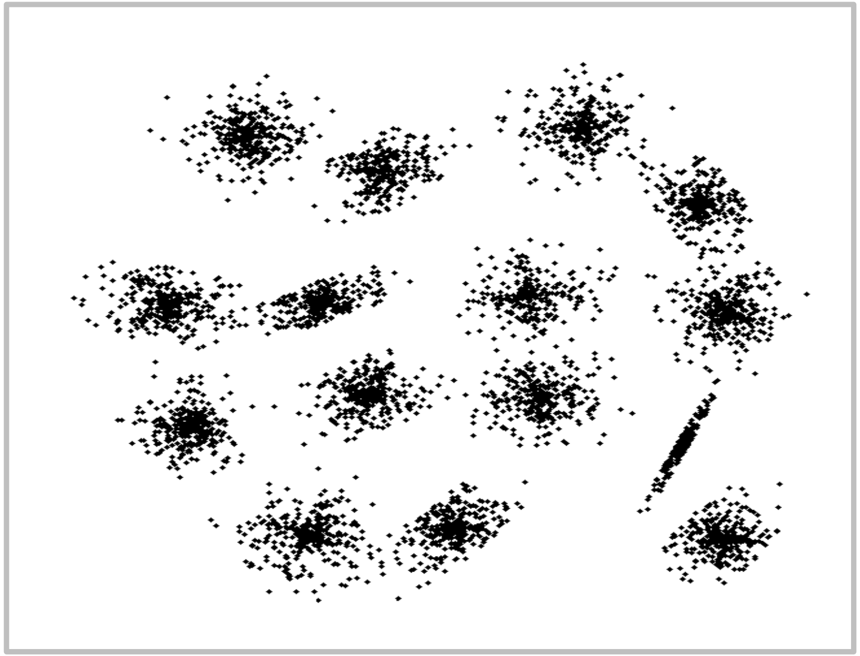
- Two approaches:
  - *Cluster analysis*
    - For identifying homogenous subgroups of samples
  - *Dimensionality Reduction*
    - For finding a low-dimensional representation to characterize and visualize the data

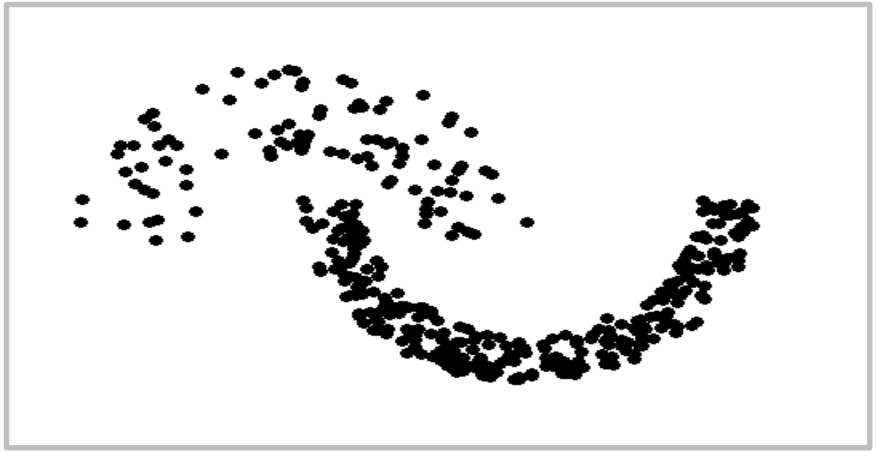
Cluster Analysis & K-means

Clustering

- *Clustering:* a set of methods for finding subgroups within the data set
  - Observations should share common characteristics within the same subgroup, but differ across subgroups
  - Groupings are determined from the data itself – differs from classification

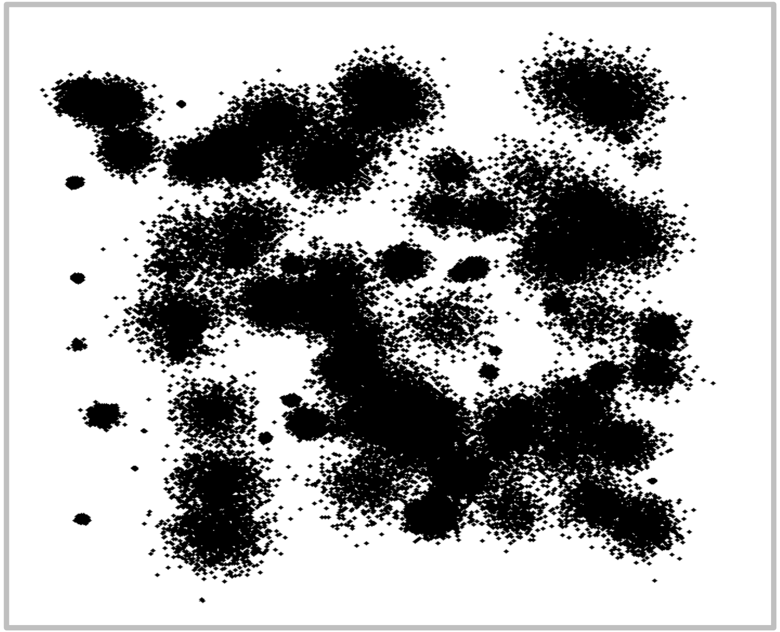
Clustering

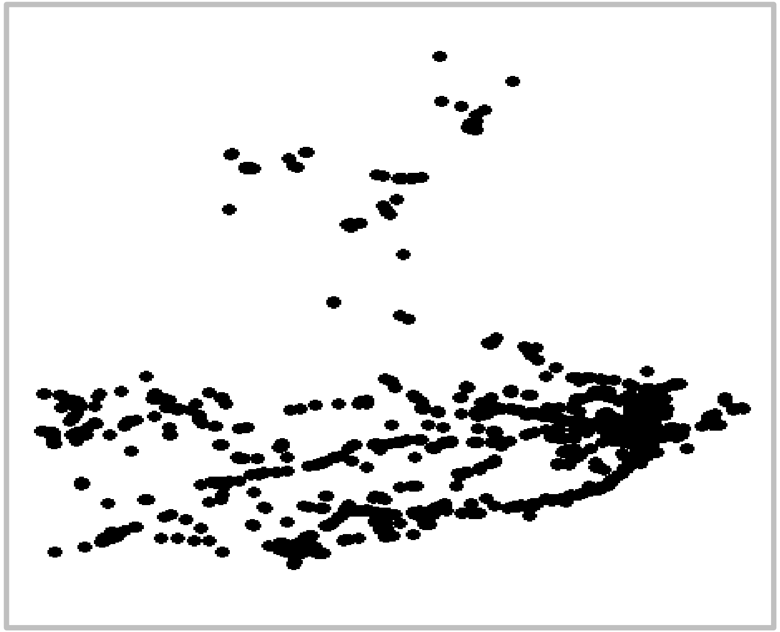




Data sets from: http://cs.joensuu.fi/sipu/datasets/

Clustering





Data sets from: http://cs.joensuu.fi/sipu/datasets/

Clustering

- Types of clustering models
  - Centroid-based clusters
  - Hierarchical clustering
  - Model-based clustering
    - Each cluster is represented by a parametric distribution
    - Data set is a mixture of distributions
  - Hard vs. soft/fuzzy clustering
    - Hard: Observations divided into distinct clusters
    - Soft: Observations may belong to more than one cluster

K-means Clustering

- Groups data into K distinct clusters
  - Each of K clusters is defined by a centroid vector
    - Centroid vector: mean of all the samples assigned to cluster
  - Each observation assigned to a single cluster (nearest centroid)
  - Requires number of clusters K as input
  - "Good clustering" minimizes within-cluster variation
    - "Similarity" measured by Euclidean distance

K-means Clustering

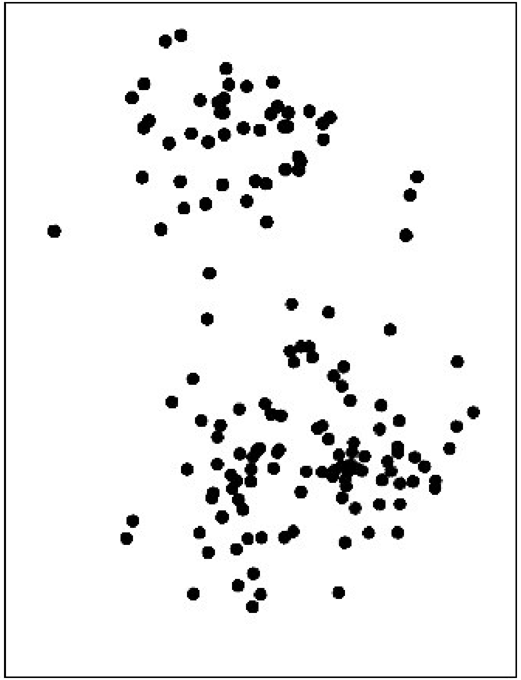


Figure 10.5 , ISL 2013*

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

K-means Clustering

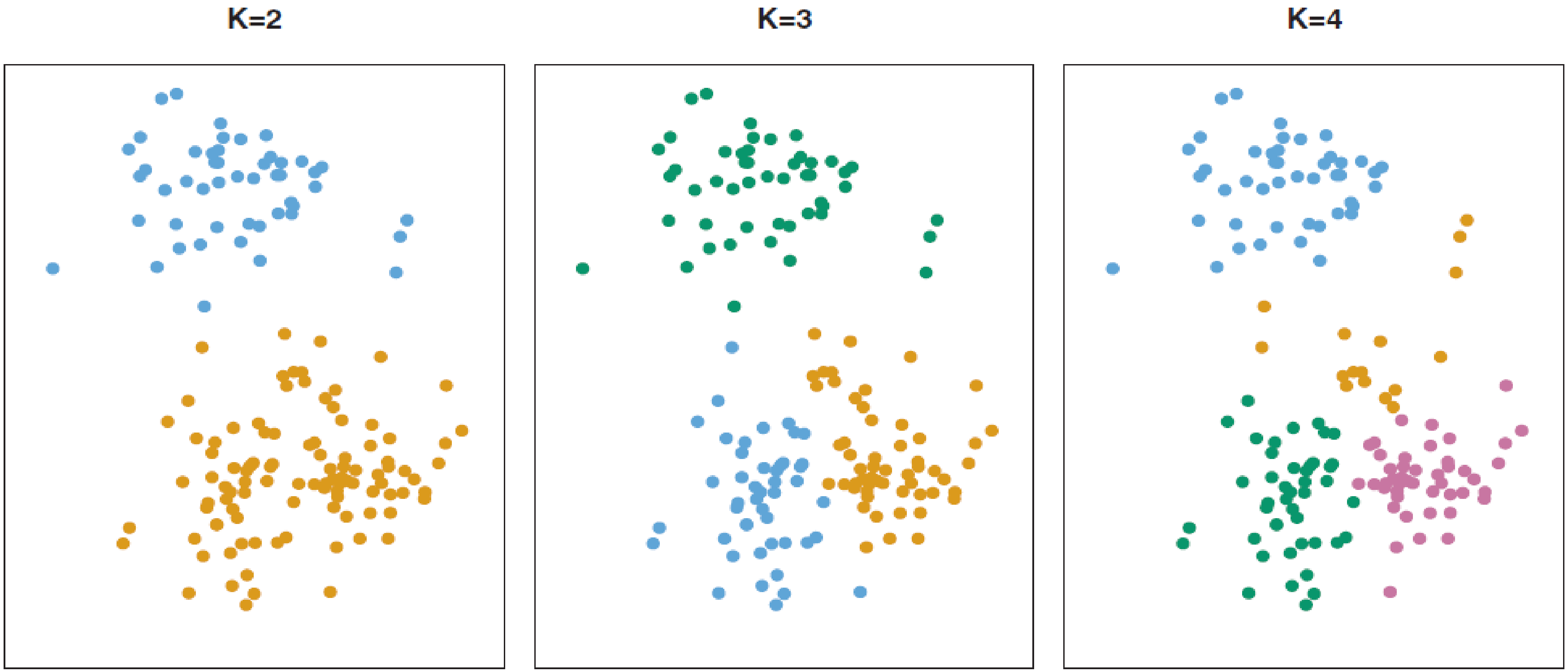


Figure 10.5 , ISL 2013

K-means Clustering

- Centroids minimize within-cluster variation

$$J = \frac{1}{n} \sum_{i=1}^{K} \sum_{x \in c_i} |x - \mu_i|^2$$

  - Centroids (cluster centers): $\mu_i = \frac{1}{|c_i|} \sum_{x \in c_i} x$

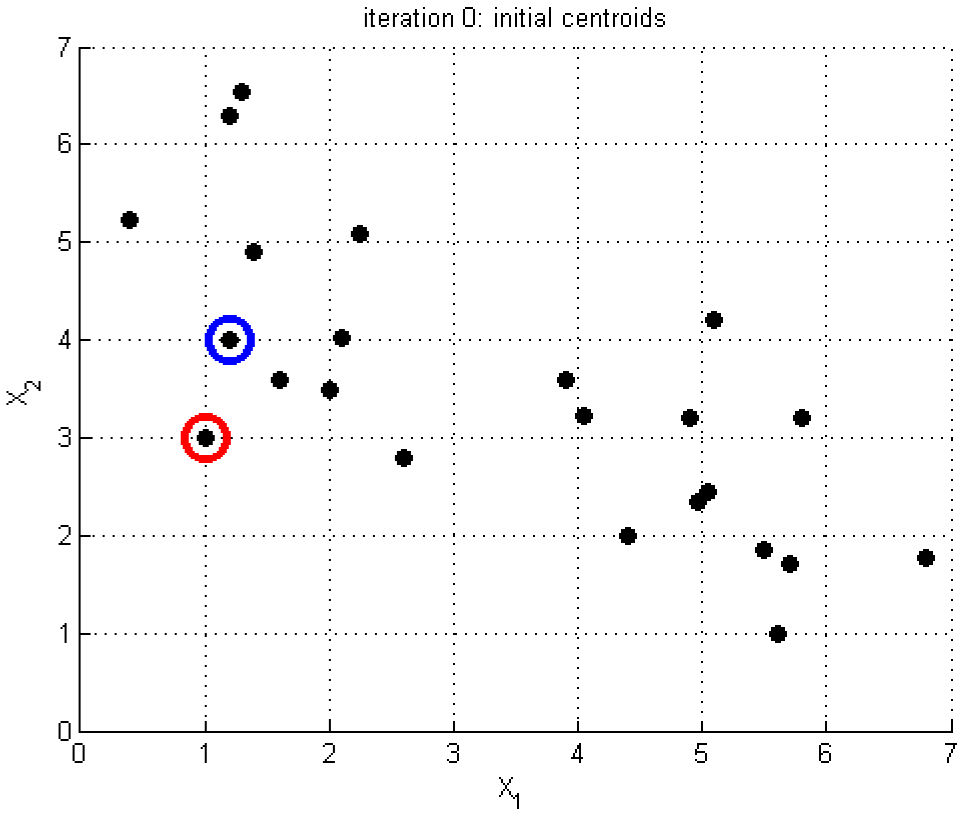
- Minimization is a combinatorial optimization problem
  - Solve for local minimum using an iterative method

K-means Algorithm*

1) Select initial set of centroids
2) Partition data by assigning each sample to cluster associated with its nearest centroid
3) Compute new centroids within each cluster
4) Repeat 2 and 3 until convergence
   – "converged" when centroids stabilize and samples do not move between clusters

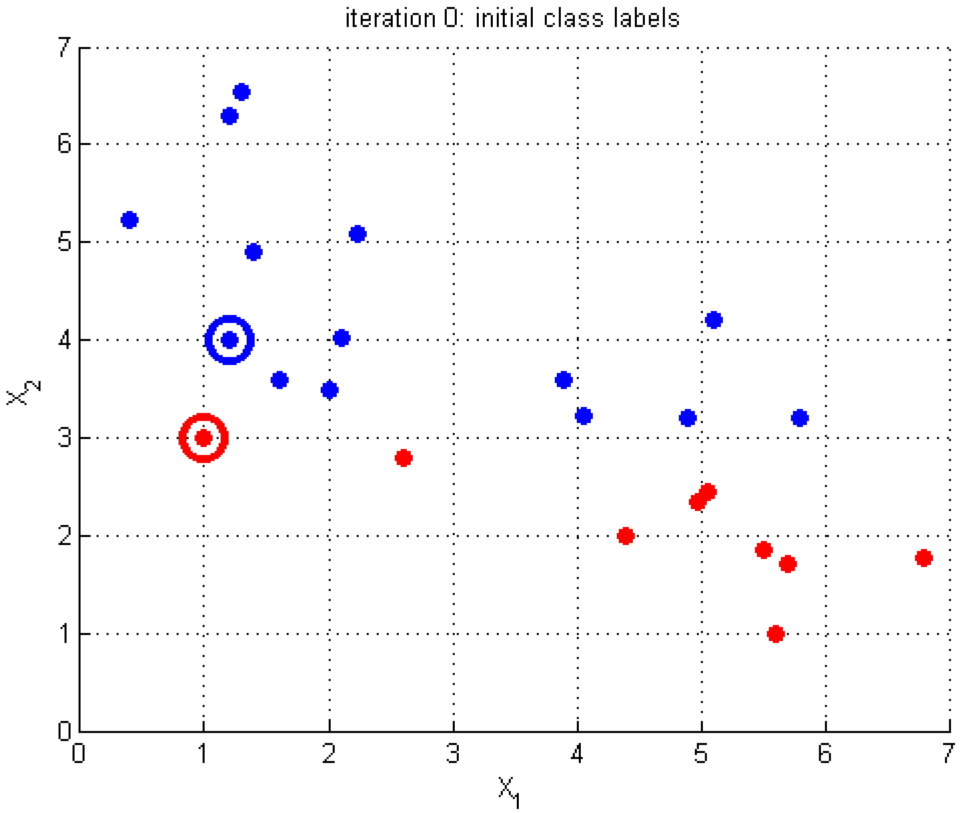
* also known as "Lloyd's algorithm"

K-means Algorithm



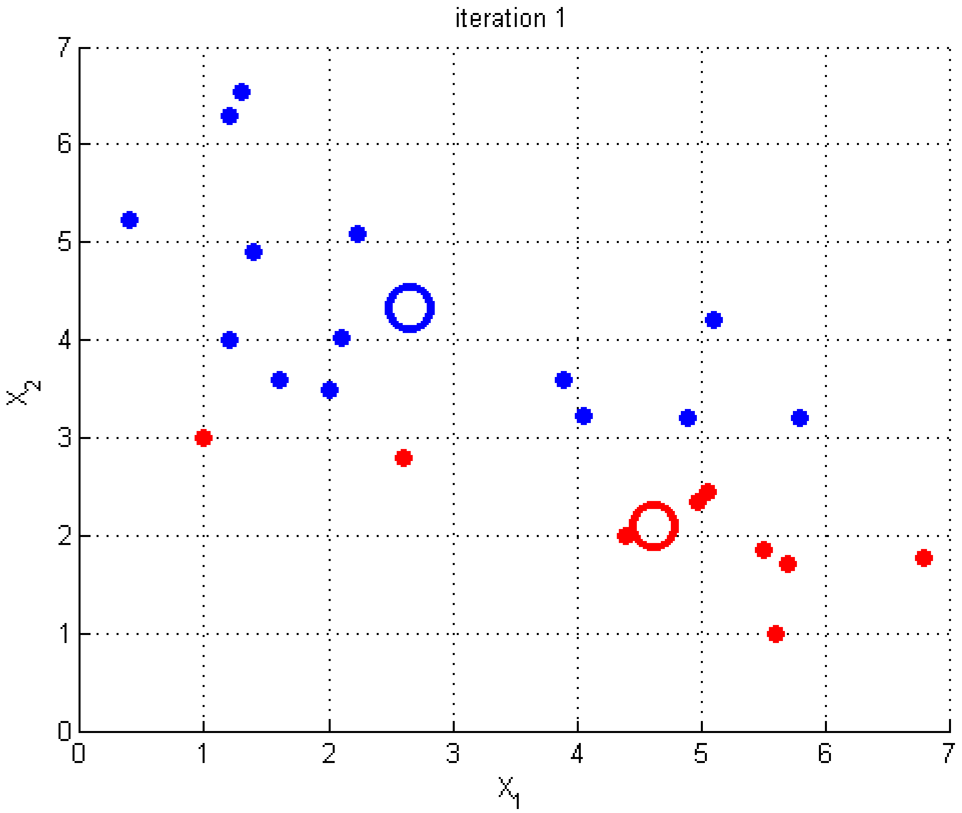
Pick initial centroids

K-means Algorithm



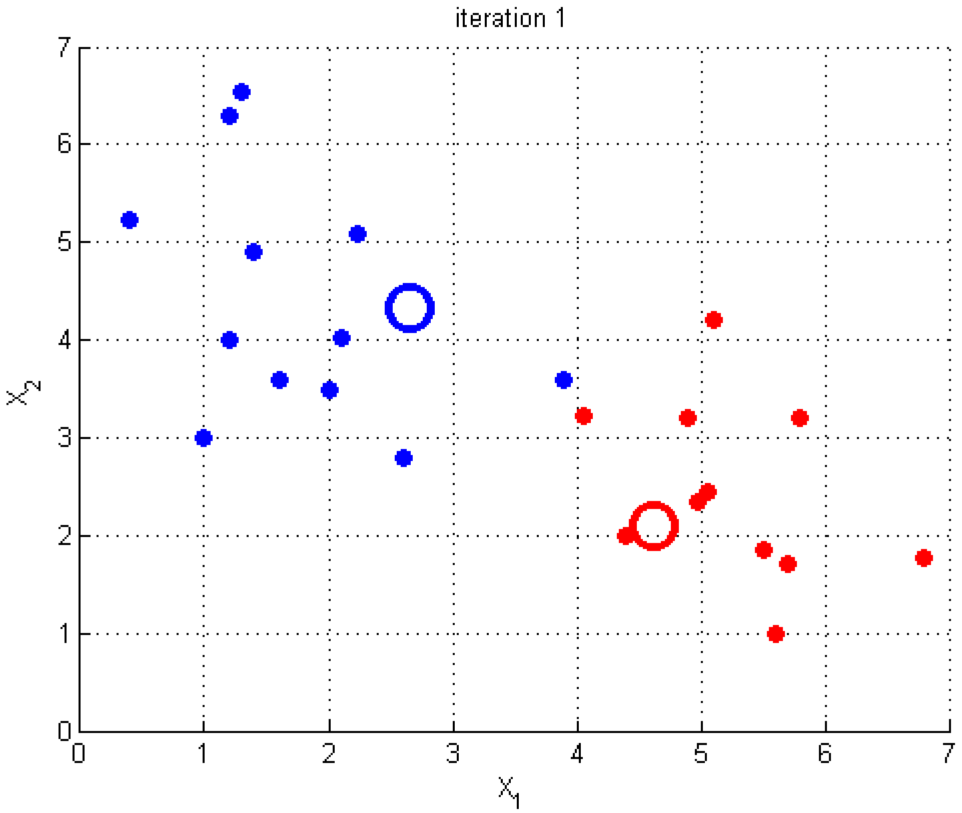
Pick initial centroids
Assign initial clusters

K-means Algorithm



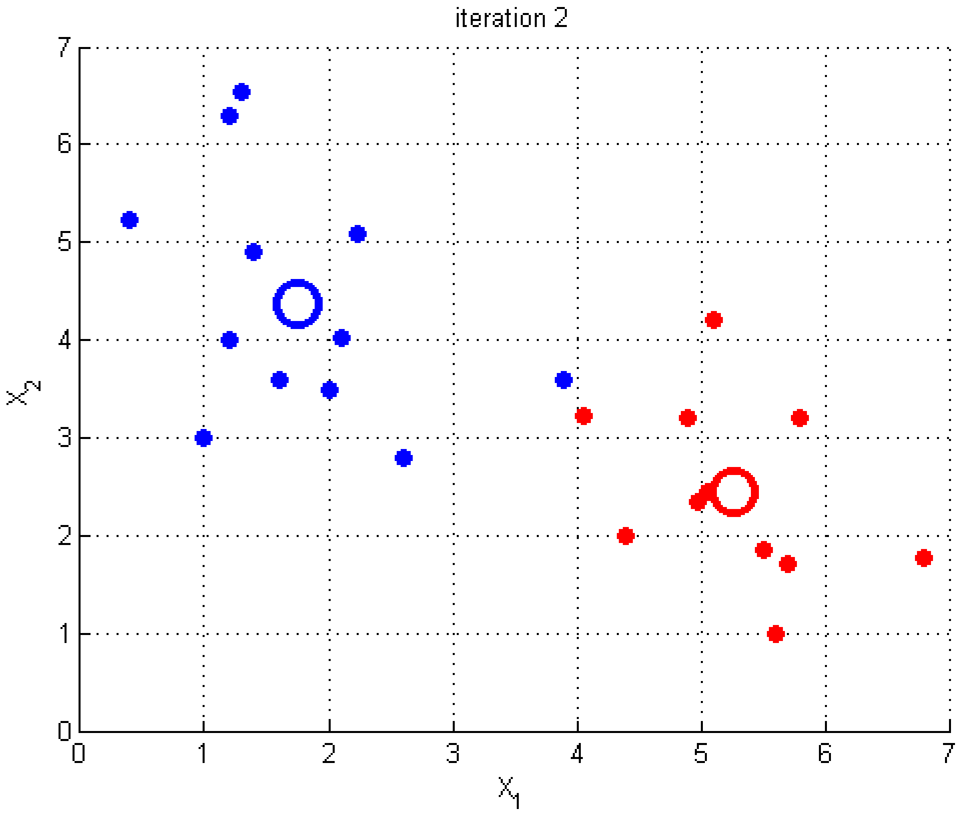
Pick initial centroids
Assign initial clusters
Update centroids

K-means Algorithm



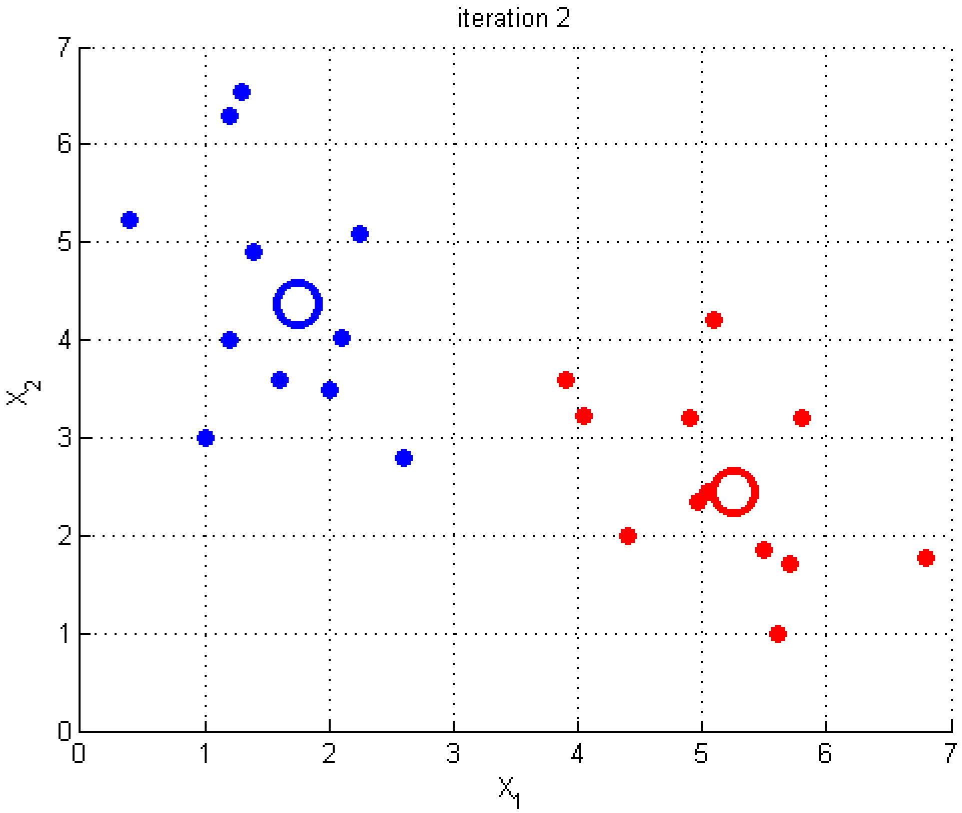
Pick initial centroids
Assign initial clusters
Update centroids
Reassign clusters

K-means Algorithm



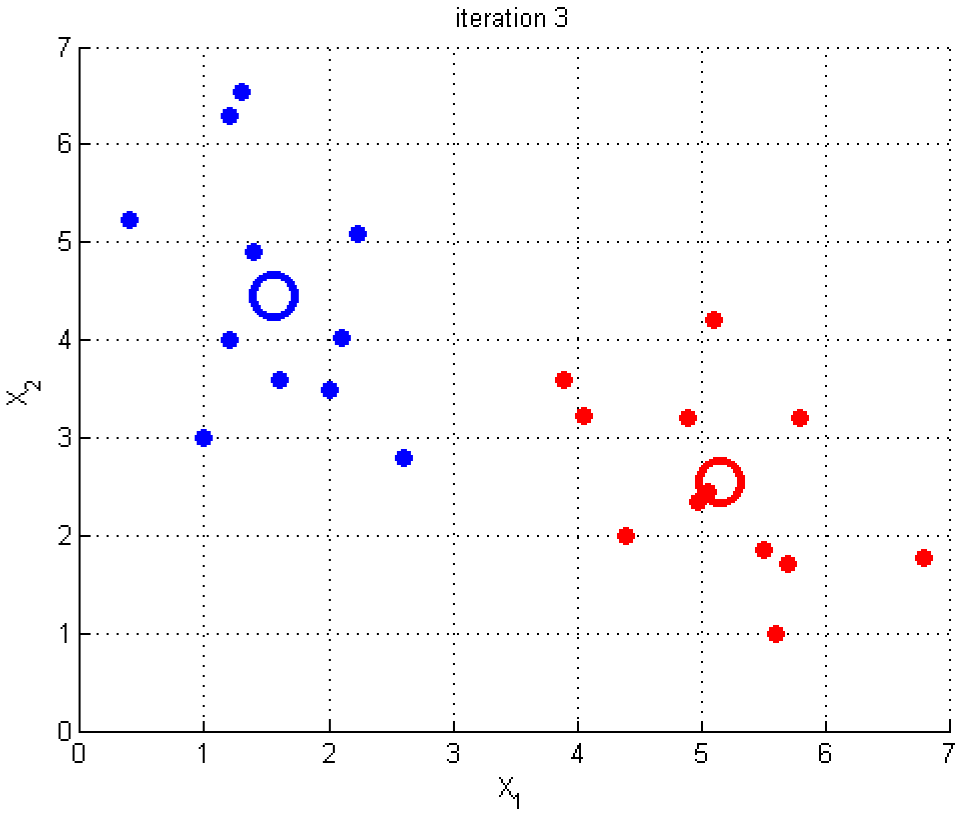
Pick initial centroids
Assign initial clusters
Update centroids
Reassign clusters
Update centroids

K-means Algorithm



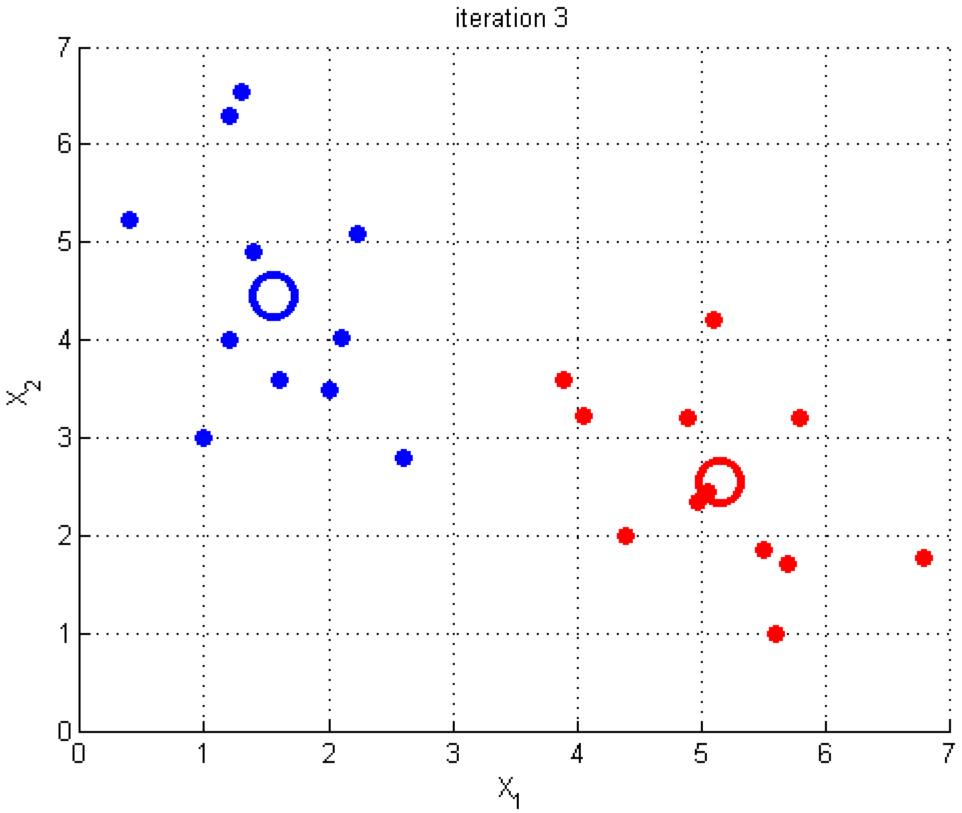
Pick initial centroids
Assign initial clusters
Update centroids
Reassign clusters
Update centroids
Reassign clusters

K-means Algorithm



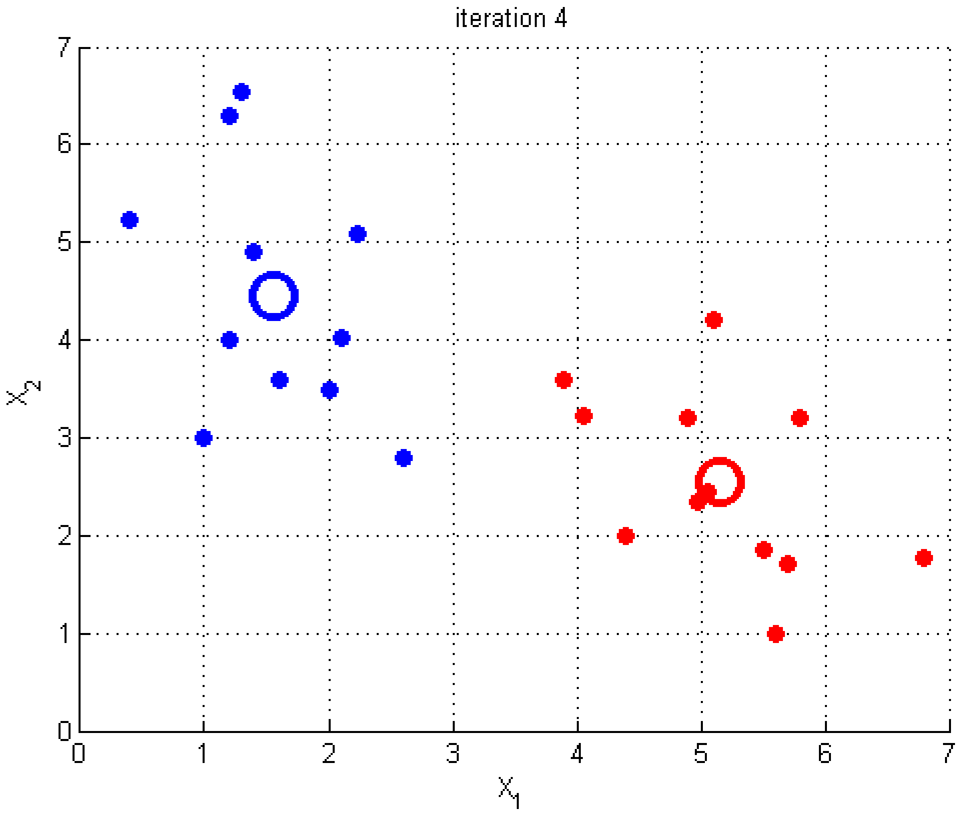
Pick initial centroids
Assign initial clusters
Update centroids
Reassign clusters
Update centroids
Reassign clusters
Update centroids

K-means Algorithm



Pick initial centroids
Assign initial clusters
Update centroids
Reassign clusters
Update centroids
Reassign clusters
Update centroids
Reassign clusters

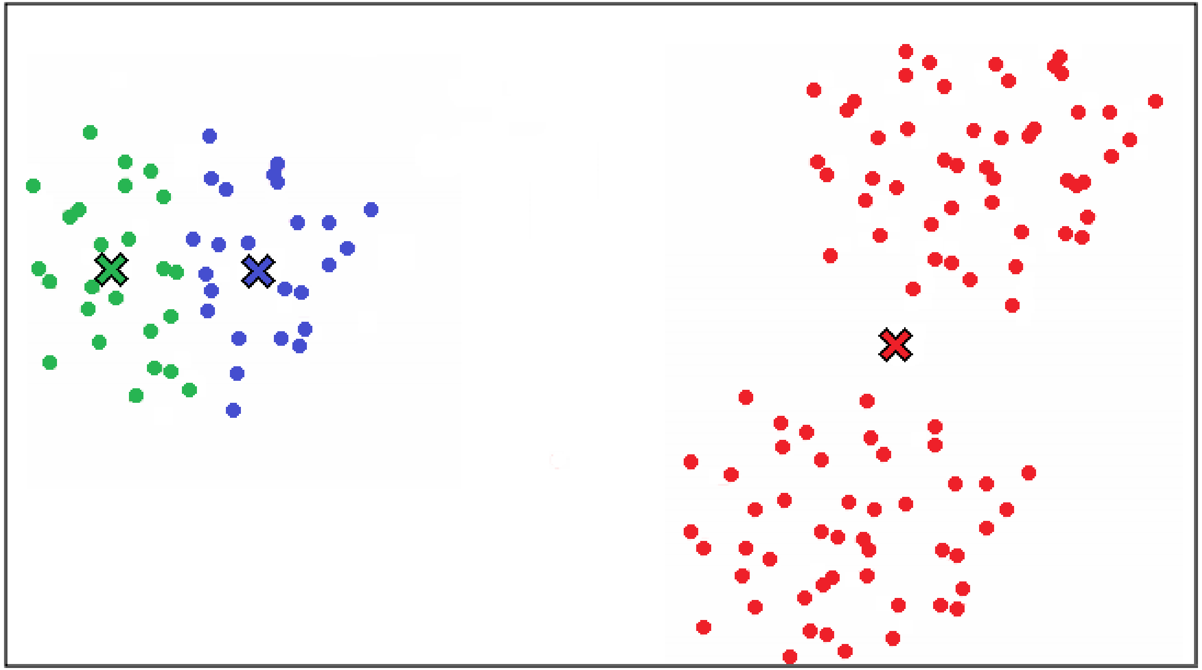
K-means Algorithm



Pick initial centroids
Assign initial clusters
Update centroids
Reassign clusters
Update centroids
Reassign clusters
Update centroids
Reassign clusters
Converged

K-means Algorithm

- Poor initialization leads to poor clustering



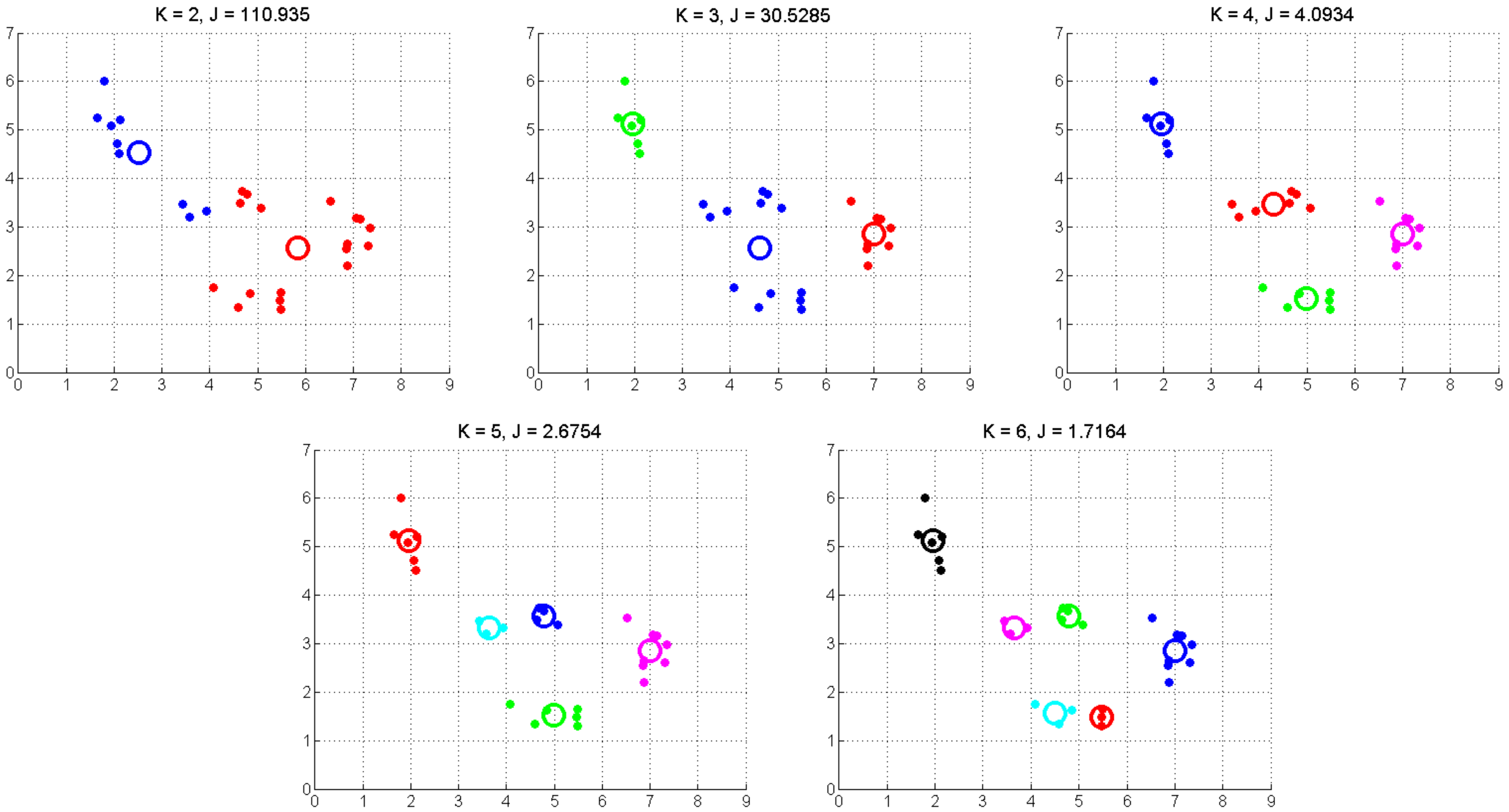
ICME

Initializing Centroids

- Random selection of K samples
- Random partition of data
- Select K points that are mutually “far apart”
- Domain-knowledge
  - Identify samples that we expect to belong to different clusters
- Initialize using results from another clustering method

How many clusters?

- K-means requires we provide K (# clusters) as input
  - We may have domain-knowledge to inform choice of K
  - Otherwise, choice of K is determined from data


K = 2, J = 110.935
K = 3, J = 30.5285
K = 4, J = 4.0934
K = 5, J = 2.6754
K = 6, J = 1.7164

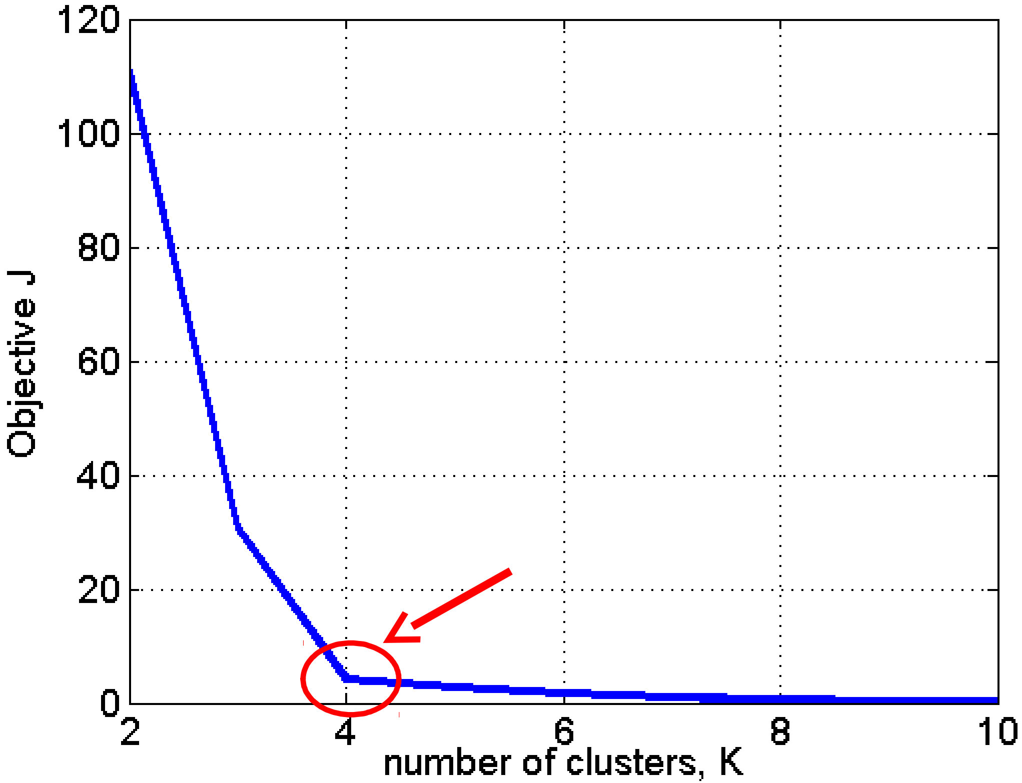
How many clusters?

- Can't just pick value of K that minimizes objective J
  - J decreases monotonically with increasing K

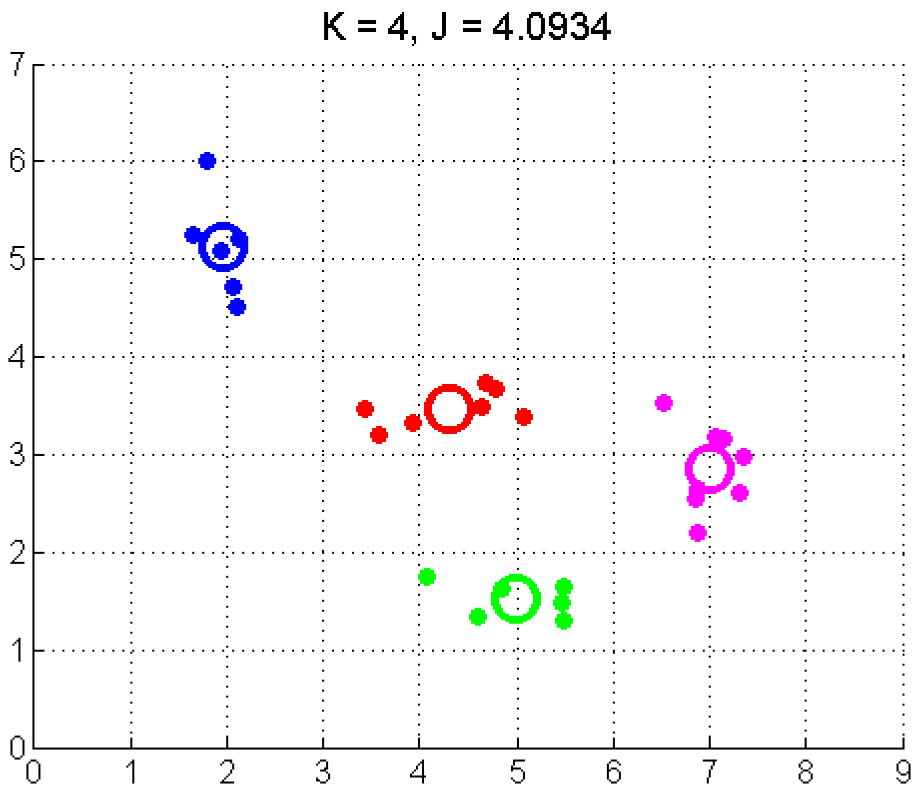
- Heuristic method:
  - For each candidate value K,
    - Compute K-means clustering M times, find minimum objective J_K
  - Find "elbow" in objective curve (K vs J_K)

ICME

How many clusters?



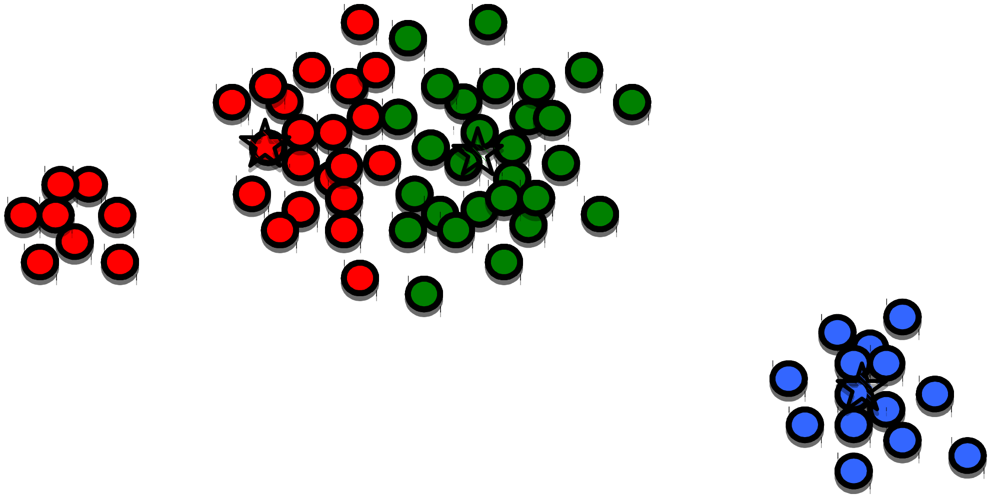
How many clusters?



K-means algorithm

- Advantages
  - Easy to implement
  - Often converges in a small number of iterations
  - Can be applied on data with a large number of features
- Disadvantages
  - K is input parameter
  - Iterative algorithm returns local minimum*

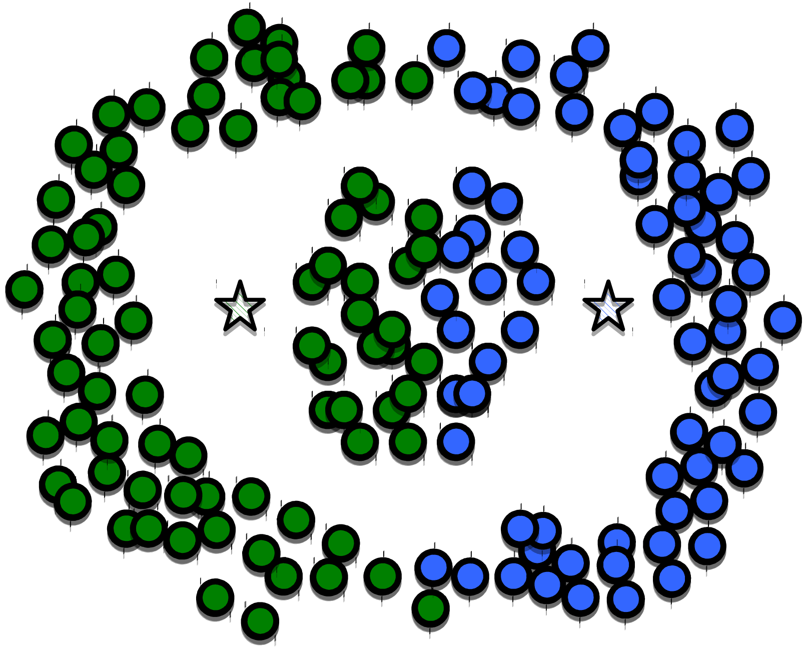
K-means algorithm



K-means algorithm

- Advantages
  - Easy to implement
  - Often converges in a small number of iterations
  - Can be applied on data with a large number of features
- Disadvantages
  - K is input parameter
  - Iterative algorithm returns local minimum*
  - Assumes all clusters spherical and approximately the same size*

K-means algorithm

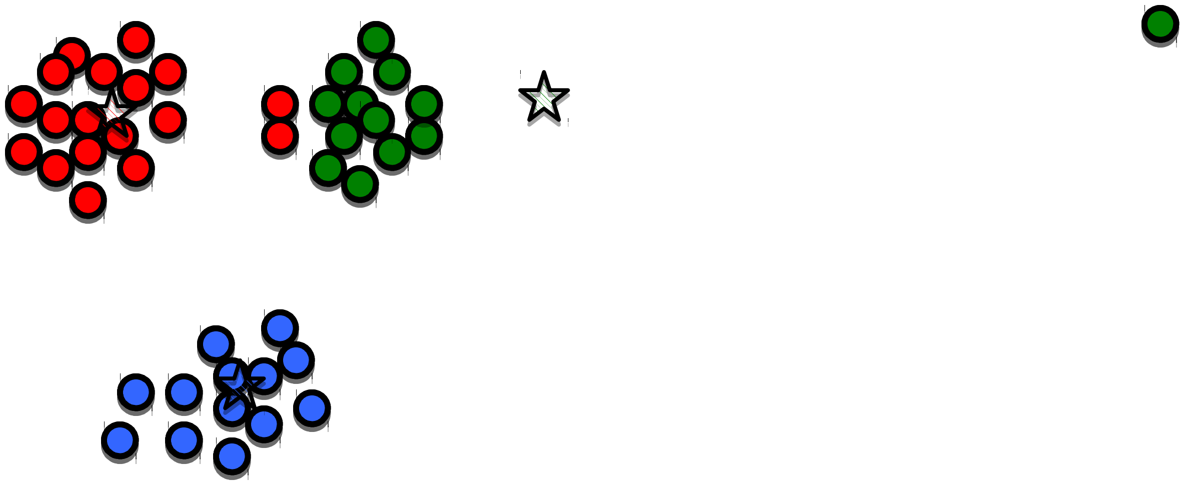


K-means algorithm

- Advantages
  - Easy to implement
  - Often converges in a small number of iterations
  - Can be applied on data with a large number of features
- Disadvantages
  - K is input parameter
  - Iterative algorithm returns local minimum*
  - Assumes all clusters spherical and approximately the same size*
  - Sensitive to outliers*

ICME

K-means algorithm



K-means algorithm

- Advantages
  - Easy to implement
  - Often converges in a small number of iterations
  - Can be applied on data with a large number of features
- Disadvantages
  - K is input parameter
  - Iterative algorithm returns local minimum*
  - Assumes all clusters spherical and approximately the same size*
  - Sensitive to outliers*
  - ***some disadvantages handled by variants of K-means**

ICME

Example:
Image Segmentation/Compression

ICME

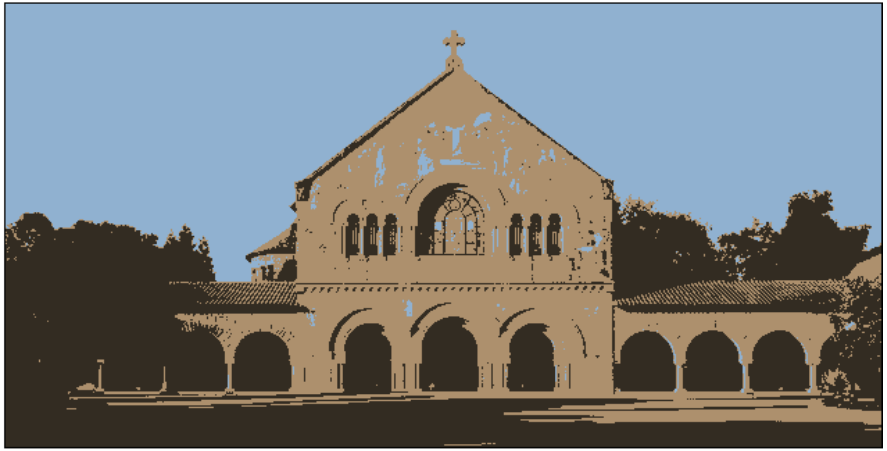
Image Segmentation/Compression

- Image -> pixels -> RGB vectors (colors)
- Apply K means to collection of RGB vectors
  - One RGB vector per pixel
  - Clusters represent similar colors
- Replace each pixel with its associated centroid
  - Results in image with only K different colors

ICME

Image Segmentation/Compression

K = 3



ICME

Image Segmentation/Compression

K = 4

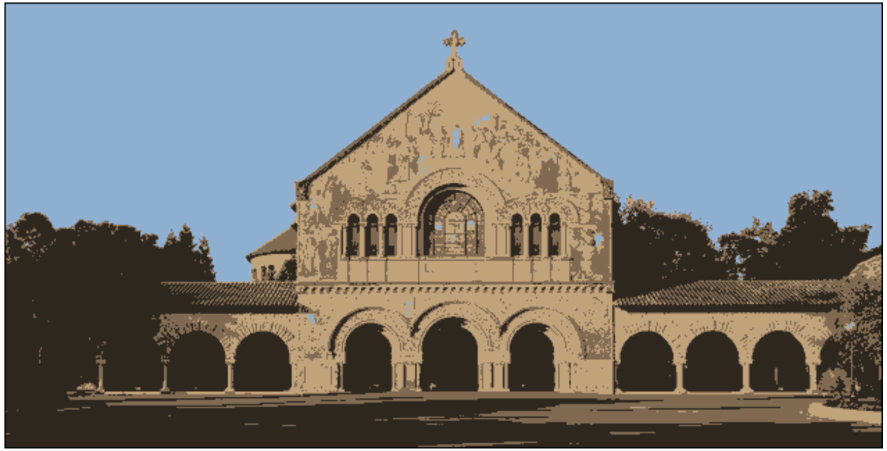


Image Segmentation/Compression

K = 5

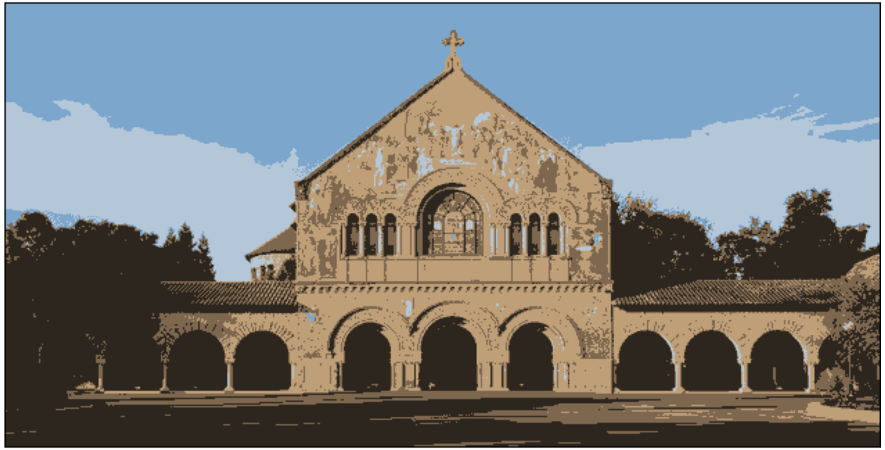


Image Segmentation/Compression

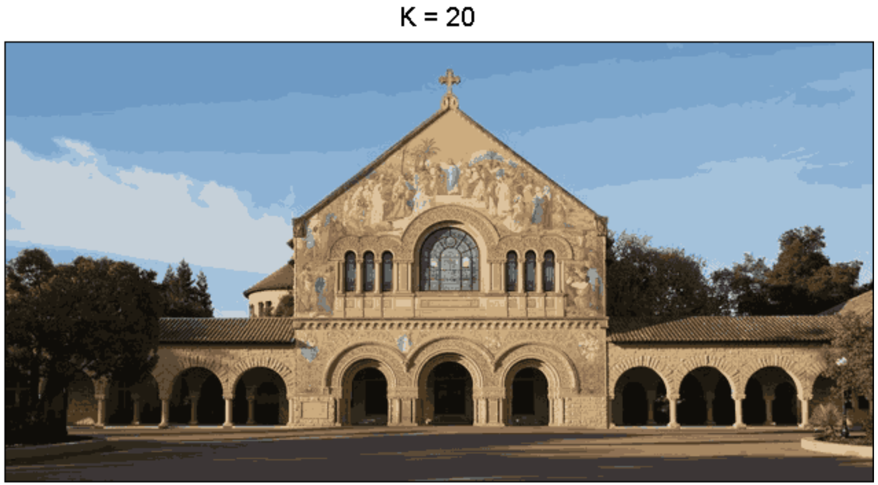
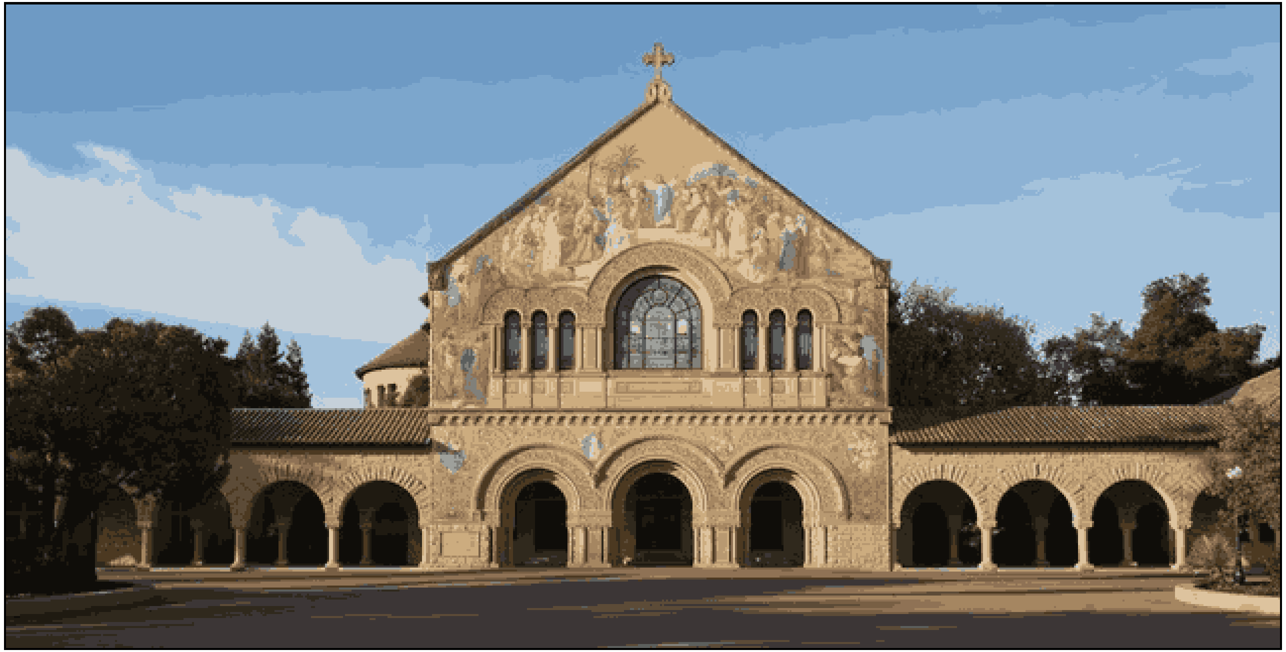


Image Segmentation/Compression

K = 20



Questions?

Introduction to Supervised Learning

Gabriel Maher
gdmaher@stanford.edu

Institute for Computational and Mathematical Engineering, Stanford University

Supervised Learning

Supervised Learning

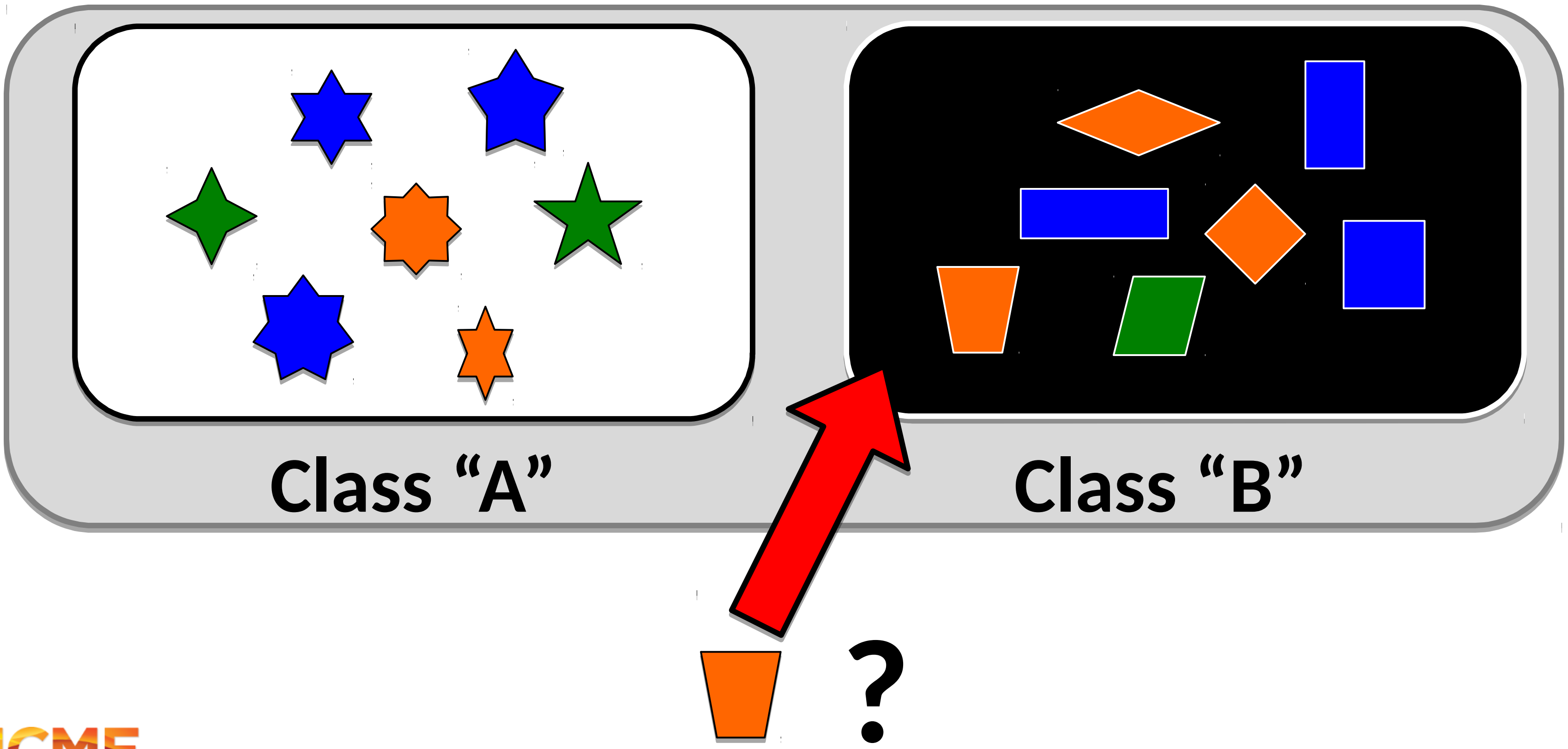
- Recall framework $Y = f(X) + \epsilon$

- Supervised learning methods:
  - "Learn by example"
  - Build a model $\hat{f}$ using set of labeled observations

$$\left(X^{(1)}, Y^{(1)}\right), \ldots, \left(X^{(n)}, Y^{(n)}\right)$$

ICME

Training Data



Training data

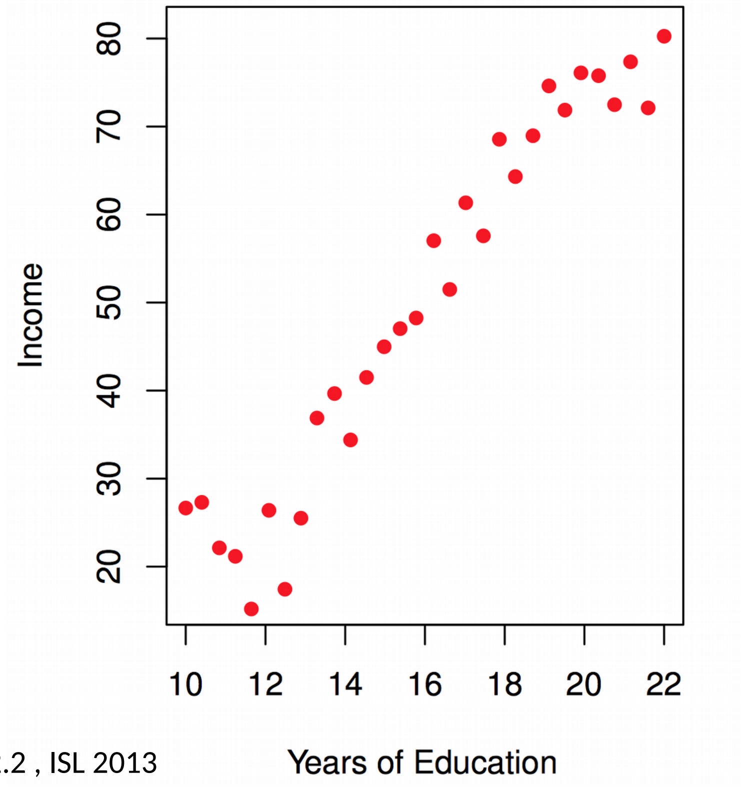


Figure 2.2 , ISL 2013

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

Supervised Learning

- Supervised learning algorithms
  - Pick "best" estimate $\hat{f}$ from among family of functions

- Example: Linear regression
  - Select best fit to *training data* within set of all linear functions

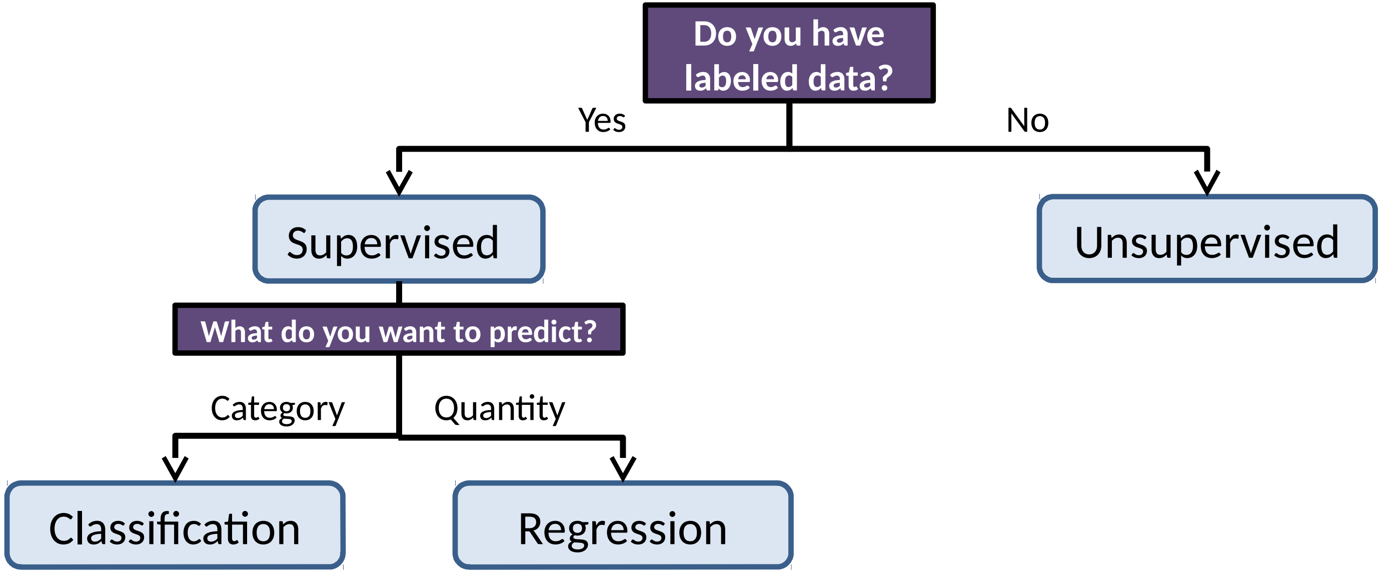
$$f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_d X_d$$

Classification and Regression

- Supervised learning problems belong to two types
  - Regression: the output Y is quantitative
  - Classification: the output Y is qualitative/categorical

ICME

Types of Algorithms



Assessing Model Accuracy

Measuring Performance: Regression

- Loss function: quantifies cost of errors

- e.g. Mean squared error (MSE)
  - Common measure of prediction accuracy in regression

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

  - Penalizes large errors more than small errors

Measuring Performance: Regression

- Our goal: build model that *generalizes*
  - We want to minimize errors on unseen "test data," not on the training data
  - e.g. predicting *future* stock prices vs. past stock prices

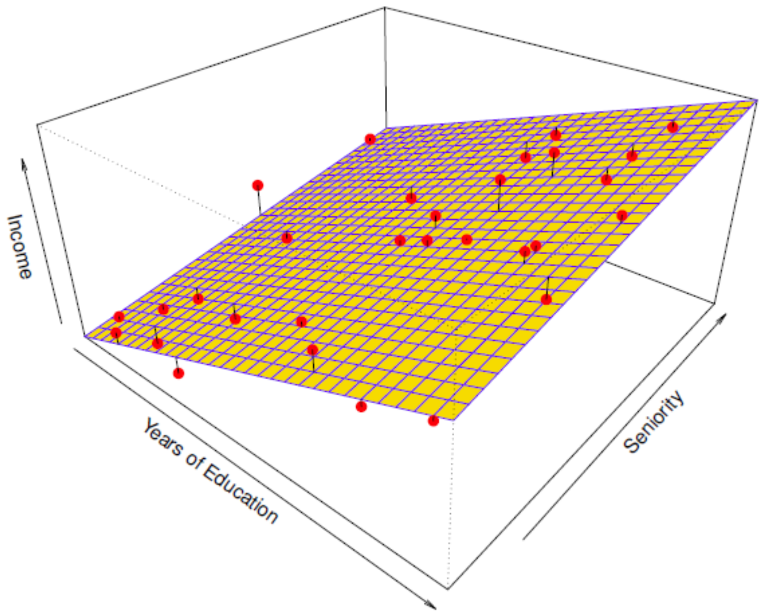
- Want to minimize *expected* loss
  - Problem: Can't just minimize loss on training data

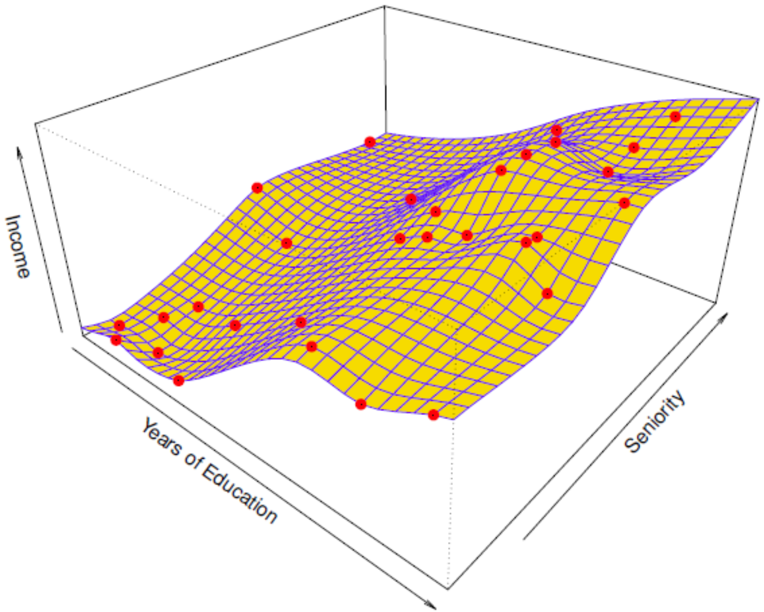
Challenge: Overfitting

- *Overfitting*: learning the random variation in the data rather than the underlying trend

- Characteristic of overfitting:
  - Good performance on previously-seen (training) data, but poor performance on new data

Challenge: Overfitting





Figures 2.4 and 2.6 , ISL 2013

Measuring Performance

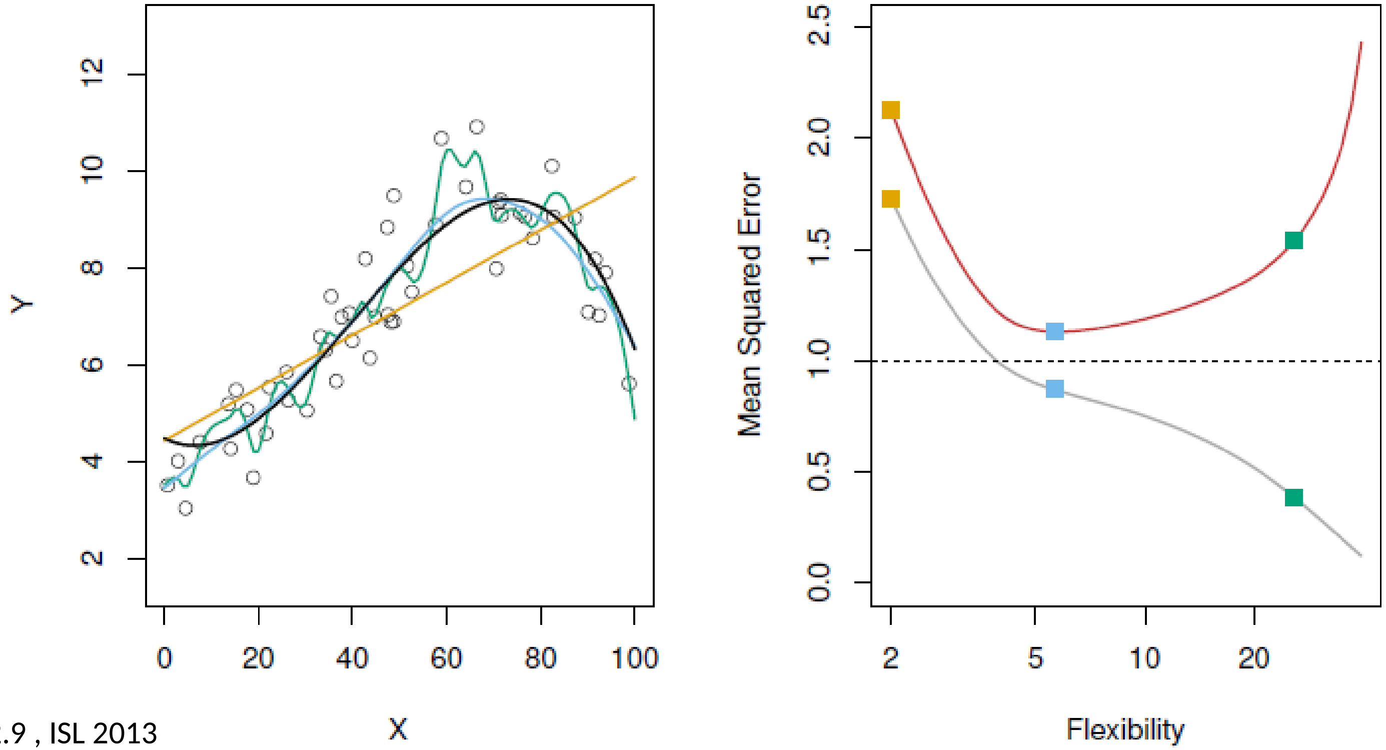


Figure 2.9 , ISL 2013

Measuring Performance

- How do we estimate test error to find a good model?

- *Cross-validation:*

a set of techniques for using the training data to estimate generalization error

Data sets

- *Training data*
  – Set of observations used to learn the model
- *Validation data*
  – Set of observations used to estimate error for parameter-tuning or model selection
- *Test data*
  – Set of observations used to measure performance on unseen data
  – These data are **not** available to the algorithm during learning process

Trade-off: Bias vs. Variance

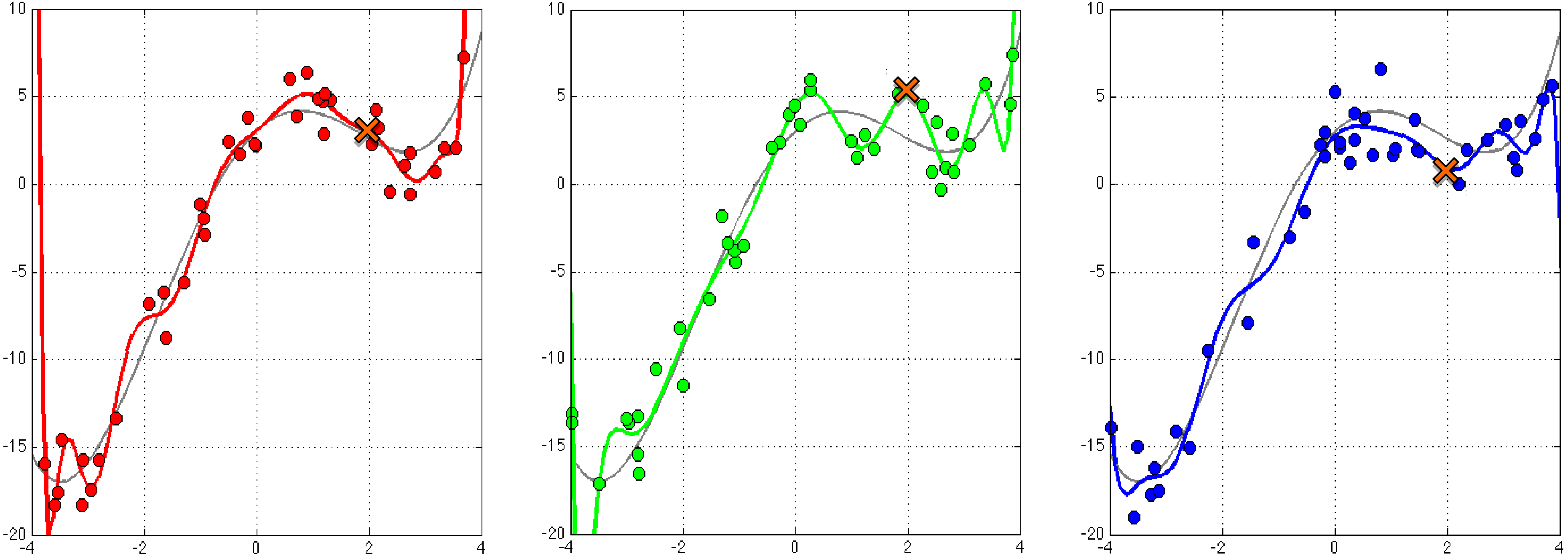
- U-shaped test error due to two competing properties of learning methods:

$$\mathbb{E}\left[\text{test error}\right] = var(\hat{f}) + bias(\hat{f})^2 + var(\epsilon)$$

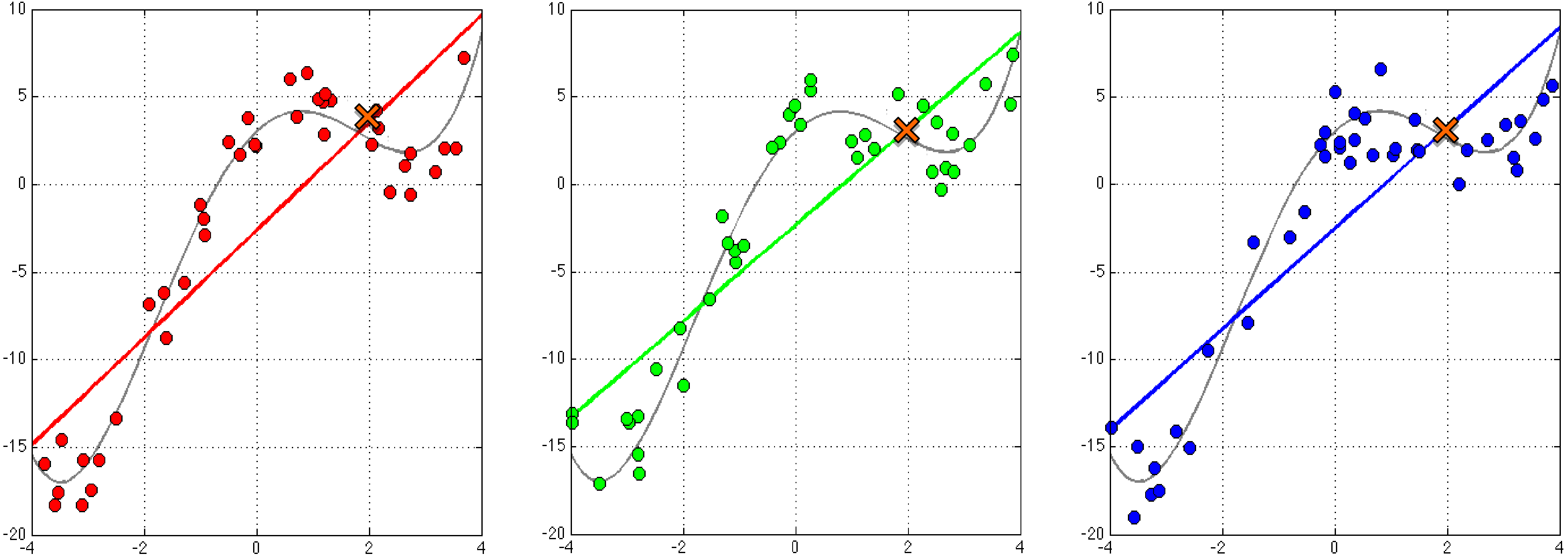
  - $var(\hat{f})$: *variance* of estimator
  - $bias(\hat{f})$: *bias* of estimator

ICME

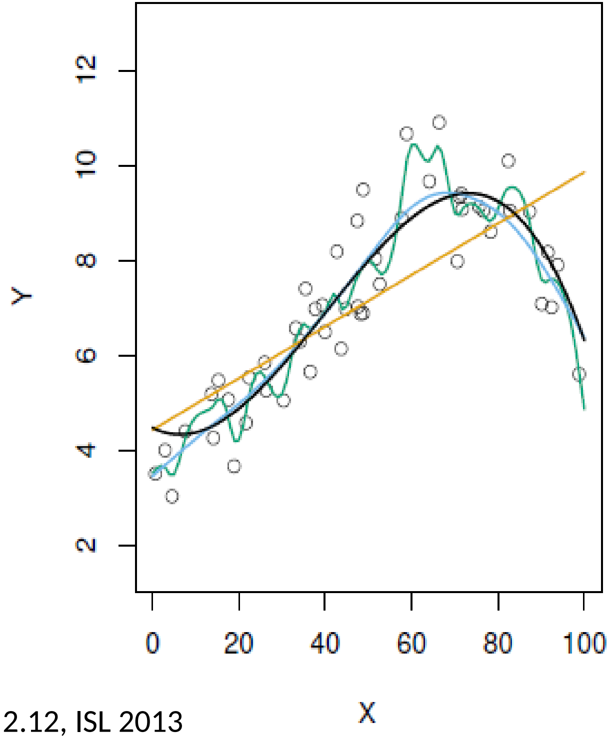
Trade-off: Bias vs. Variance

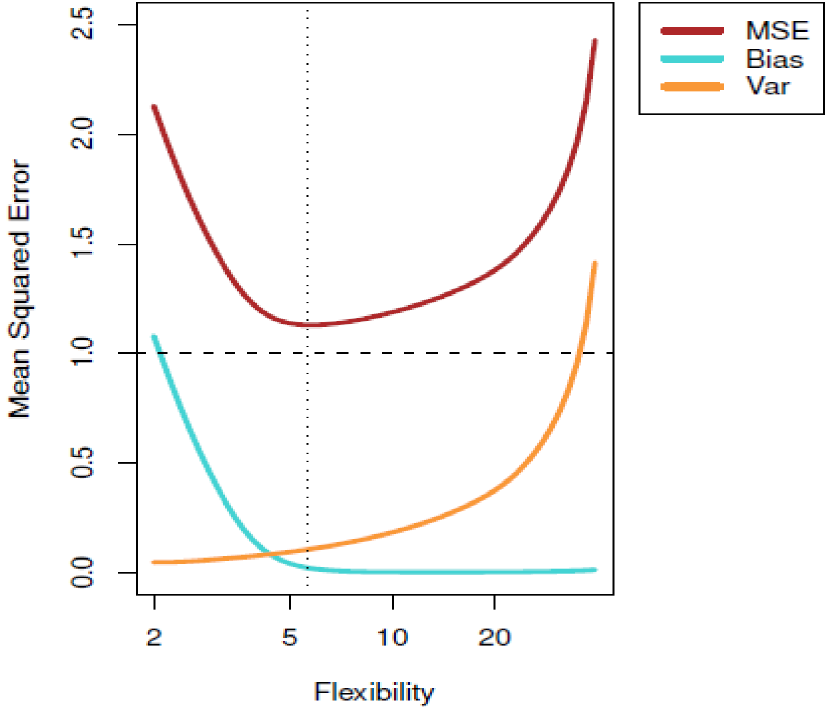


Trade-off: Bias vs. Variance



Trade-off: Bias vs. Variance





Figures 2.9, 2.12, ISL 2013

ICME

Trade-off: Bias vs. Variance

- Flexible method
  - Can achieve closer fit to underlying system (low bias),
  - But risks learning model that is too dependent on training data (high variance)

- Simpler method
  - May not accurately model underlying system (high bias),
  - But less dependent on training data (low variance)

- Trade-off
  - Easy to achieve low variance/high bias *or* high variance/low bias,
  - But hard to achieve low variance *and* bias

Regression: Linear Regression

Linear Regression

- *Linear Regression*: a simple supervised learning method, used to predict quantitative output values
  - Many machine learning methods are generalizations of linear regression
  - Example to illustrate key concepts in supervised learning

ICME

Simple Linear Regression

- Output Y and single predictor X with linear relationship between X and Y:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

- Model parameters:

$$\beta_0 \text{ intercept}$$
$$\beta_1 \text{ slope}$$

ICME

Simple Linear Regression

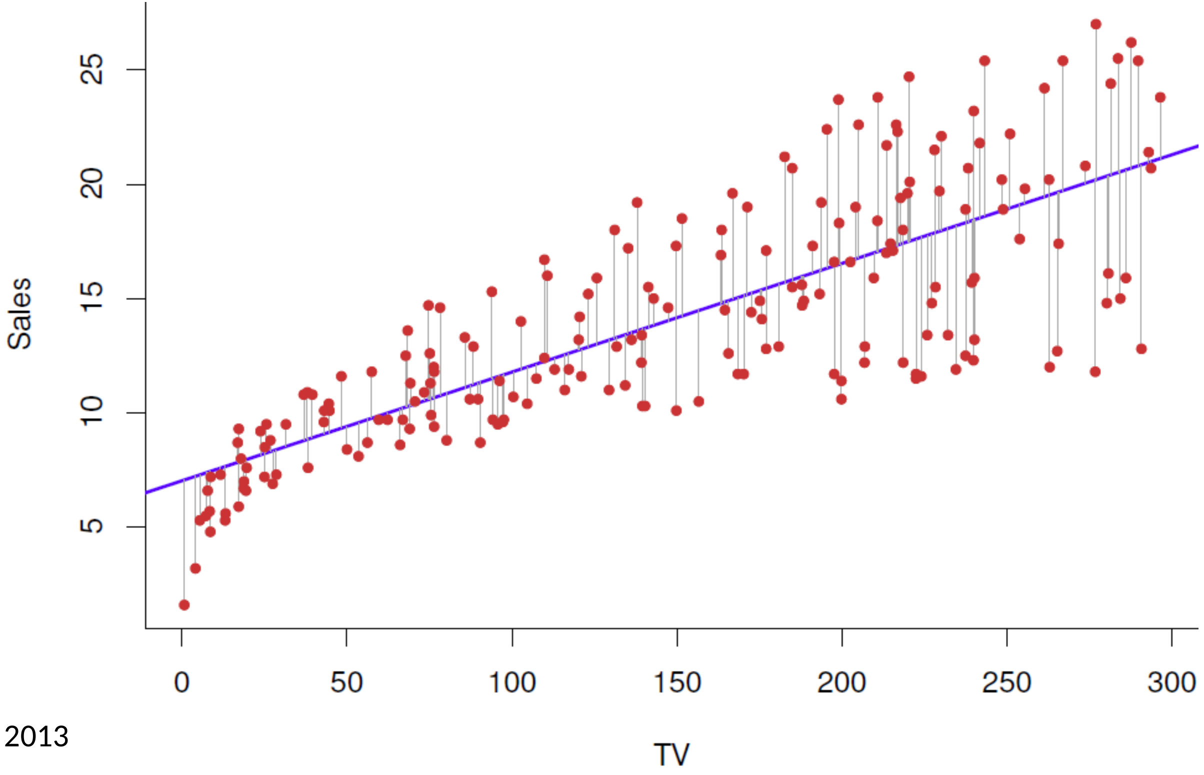


Figure 3.1 , ISL 2013

Simple Linear Regression

- β_0 and β_1 are unknown -> estimate their values using training data

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

- Pick $\hat{\beta}_0, \hat{\beta}_1$ such that the model is a "good fit" to training data

$$Y^{(i)} \approx \hat{\beta}_0 + \hat{\beta}_1 X^{(i)}, \quad i = 1, \ldots, n$$

ICME

Simple Linear Regression

- How do we estimate the coefficients ("fit the model")?
- What makes model a "good fit" to the data?

Least Squares

- Typically, fit of the model to the data is measured using *least squares*
- Mean squared error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y^{(i)} - \hat{Y}^{(i)}\right)^2$$

Least Squares

- "Fit model" - coefficients $\hat{\beta}_0, \hat{\beta}_1$ minimize MSE:

$$\min_{(\hat{\beta}_0, \hat{\beta}_1)} \left[\frac{1}{n} \sum_{i=1}^{n} \left(Y^{(i)} - \left(\hat{\beta}_0 + \hat{\beta}_1 X^{(i)} \right) \right)^2 \right]$$

- Explicit formulas for solution:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} \left(X^{(i)} - \bar{X} \right) \left(Y^{(i)} - \bar{Y} \right)}{\sum_{i=1}^{n} \left(X^{(i)} - \bar{X} \right)^2} \qquad \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$$

ICME

Simple Linear Regression

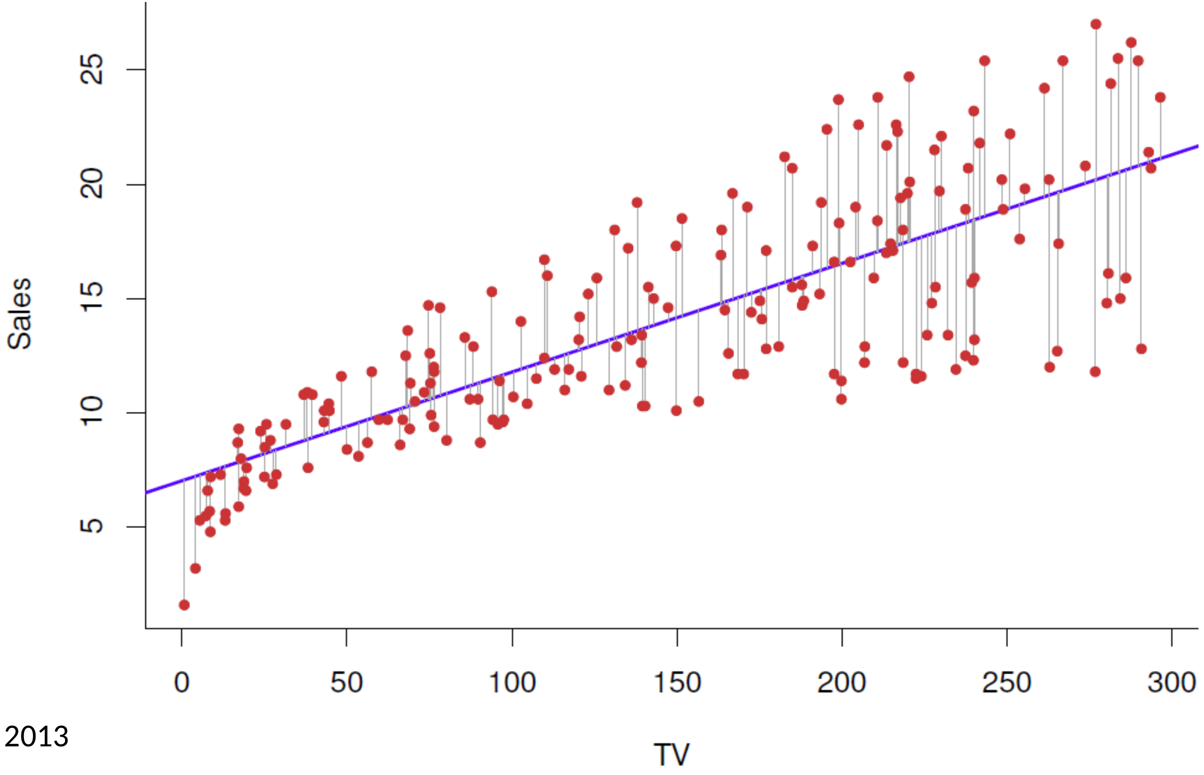


Figure 3.1 , ISL 2013

Multiple Linear Regression

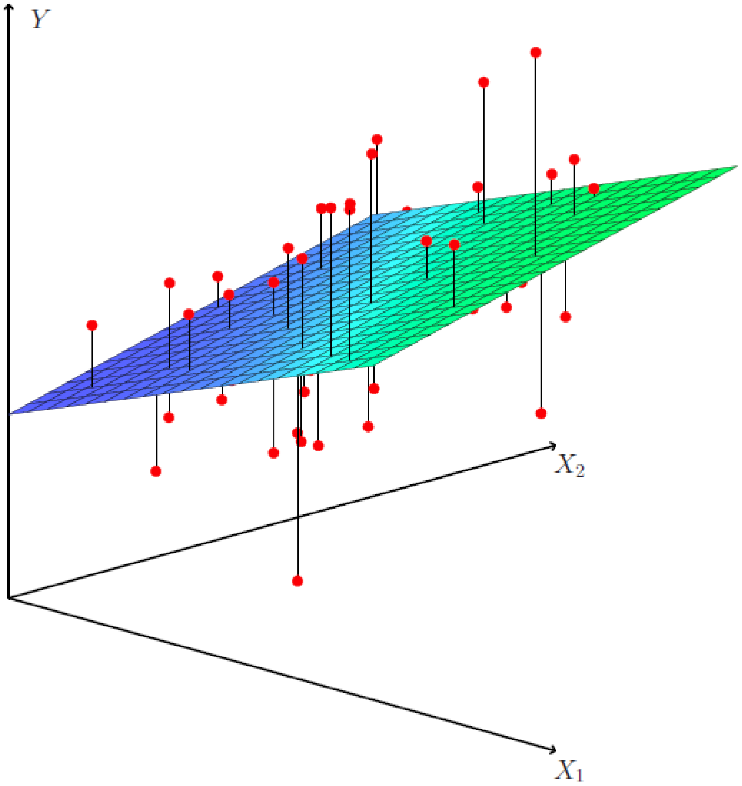


Figure 3.4 , ISL 2013

Multiple Linear Regression

- Multiple linear regression: more than one predictor variable used to predict response

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_d X_d + \epsilon$$

Least Squares

- Solve for coefficient estimates ("fit model") using least squares fit on training data:

$$\hat{\beta} = \arg\min_{\beta} \|Y - X^T\beta\|^2$$

$$X = \begin{bmatrix} 1 & X^{(1)^T} \\ & \dots \\ 1 & X^{(n)^T} \end{bmatrix} \qquad \hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \dots \\ \hat{\beta}_d \end{bmatrix}$$

- Use normal equations to solve for $\hat{\beta}$:

$$Y = \begin{bmatrix} Y^{(1)} \\ \dots \\ Y^{(n)} \end{bmatrix}$$

$$X^T X\hat{\beta} = X^T Y \quad \rightarrow \quad \hat{\beta} = \left(X^T X\right)^{-1} X^T Y$$

Linear Regression

- Advantages:
  - Simple model
  - Interpretable coefficients
  - Can obtain good results with small data sets
  - Many variations/extensions

- Disadvantages:
  - Model may be too simple to make accurate predictions over large range of values
  - Poor extrapolation
  - Sensitive to outliers in data – least squares error

ICME

Questions?

Classification: Logistic Regression

Classification

- Regression – predicting quantitative response Y
  - For some applications, response variable is qualitative or categorical

- Classification: predicting a qualitative response
  - Assign each observation to a class / category
  - e.g. K-nearest neighbor classifier from Lecture 1

ICME

Classification and Regression

- Classification and regression are closely related
- Classification as regression:
  - Predict the probability an observation belongs to each class, assign to class with highest probability

Logistic Regression

- *Binary classification*: Y takes on two values ("0" or "1") corresponding to two classes

- Logistic regression models binary classification as

$$Pr\left(Y \text{ belongs to class } 1 \mid X\right)$$

  - Threshold to obtain class decisions
  - Modified Linear regression for probabilities in [0, 1]

Logistic Regression

- Logistic (sigmoid) function bounds output $Y \in [0, 1]$
- Logistic function $\sigma(z) = \frac{e^z}{1+e^z} \left(= \frac{1}{1+e^{-z}} \right)$
  - S-shaped curve
  - Always takes values in (0, 1) -> valid probabilities
- Logistic Regression model

$$Pr(Y = 1 \mid X) = \sigma(\beta_0 + \beta_1 X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

ICME

Logistic Regression

- Model parameters β_0 and β_1 must be estimated using the training data
  - For linear regression, use *least-squares*
- Fit model parameters using different cost function
  - Binary cross-entropy

ICME

Logistic Regression

- *Binary cross-entropy*

$$L = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - 1)\log(1 - \widehat{y}^{(i)}) - y^{(i)}\log\widehat{y}^{(i)}$$

- *Bad prediction:* $y = 1, \widehat{y} = 0 \to L = \infty$

  $y = 0, \widehat{y} = 1 \to L = \infty$

- *Good prediction:* $y = 1, \widehat{y} = 1 \to L = 0$

  $y = 0, \widehat{y} = 0 \to L = 0$

ICME

Multiple Logistic Regression

- We can extend logistic regression to the case of multiple predictor variables:

$$Pr(Y=1 \mid X) = \sigma(\beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d)$$

$$= \frac{e^{(\beta_0+\beta_1 X_1+\cdots+\beta_d X_d)}}{1+e^{(\beta_0+\beta_1 X_1+\cdots+\beta_d X_d)}}$$

ICME

Logistic Regression

- Advantages:
  - Extension of linear regression
  - No hyperparameters to tune

- Disadvantages:
  - Can not model complex decision boundaries
  - Must be formulated as binary classification problem

Summary

- *Supervised learning* – learning from examples

- *Linear regression* – simple, interpretable model for predicting quantitative response

- *Logistic regression* – regression method used to predict probabilities for binary classification
  - *Maximum likelihood method*: technique for estimating parameter values

Questions?