

# Unsupervised Learning

## PCA and K-means Clustering

STAT GR5206 *Statistical Computing & Introduction to Data Science*

Gabriel Young  
Columbia University

June 28, 2017

- Final exam is due Thursday by 11:59pm.
- No class tomorrow (Thursday).
- I will post some more lecture notes by Tomorrow (no assessment on these notes: future reference).

# Unsupervised Learning

# Supervised vs. Unsupervised Learning

## Supervised Learning

- Have access to a set of  $p$  predictors  $X_1, X_2, \dots, X_p$  and a response  $Y$  both measured on the same  $n$  observations.
- The goal is to predict  $Y$  using  $X_1, X_2, \dots, X_p$  (usually by learning  $\beta$  parameters of a model).

# Supervised vs. Unsupervised Learning

## Supervised Learning - have $Y$ s

- Have access to a set of  $p$  predictors  $X_1, X_2, \dots, X_p$  and a response  $Y$  both measured on the same  $n$  observations.
- The goal is to predict  $Y$  using  $X_1, X_2, \dots, X_p$  (usually by learning  $\beta$  parameters of a model).

## Unsupervised Learning

- *Only* have access to a set of  $p$  predictors  $X_1, X_2, \dots, X_p$  measured on  $n$  observations.
- We are not interested in prediction, because we do not have an associated response variable  $Y$ .
- The goal is to discover interesting patterns about the measurements on the predictors  $X_1, X_2, \dots, X_p$ .

# Review of Supervised Techniques

## Regression: Linear Regression

Using a set of training observations (data):  $(Y_i, X_{i1}, X_{i2}, \dots, X_{ip})$  for  $i = 1, 2, \dots, n$ , we want to estimate the model

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} + \epsilon_i,$$

with  $\epsilon \sim N(0, \sigma^2)$ . The *least squares estimator* is

$$\hat{\beta} = \arg \min_{\beta} MSE(\beta) = (X^T X)^{-1} X^T Y,$$

where  $Y = (Y_1, Y_2, \dots, Y_n)^T \in \mathbb{R}^n$ ,  $X$  = design matrix  $\in \mathbb{R}^{n \times (p+1)}$ , and  $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T \in \mathbb{R}^{p+1}$ . Prediction Model:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \dots + \hat{\beta}_p X_p.$$

# Review of Supervised Techniques

## Regression: Curve fitting by optimizing

Least squares curve fitting:

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n (Y_i - r(X_i; \theta))^2.$$

'Robust' curve fitting:

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n \psi(Y_i - r(X_i; \theta)),$$

where  $\psi$  is a 'robust' loss function,  $\theta \in \mathbb{R}^d$  is a set of unknown coefficients, and  $r$  is some function of the data and the parameters we assume describes the data.

# Review of Supervised Techniques

## Regression: Ridge Regression

Now put a penalty on the *magnitude* of the coefficient vector:

$$\tilde{\beta} = \arg \min_{\beta} MSE(\beta) + \overset{\text{tuning parameter}}{\mu} \sum_{j=1}^p \beta_j^2 = \arg \min_{\beta} MSE(\beta) + \mu \|\beta\|_2^2.$$

Closed form solution exists:  $\tilde{\beta} = (X^T X + \mu I)^{-1} X^T y$

## Regression: The LASSO - forces sparsity (some estimates go to 0)

Now put a penalty on the sum of the coefficient's absolute values:

$$\beta^\dagger = \arg \min_{\beta} MSE(\beta) + \mu \sum_{j=1}^p |\beta_j| = \arg \min_{\beta} MSE(\beta) + \mu \|\beta\|_1.$$



# Review of Supervised Techniques

## Classification: Logistic Regression

Predicting a binary response using a logistic regression model (We skipped this topic in this class).

## Classification: $K$ Nearest Neighbors (KNN)

- Estimates  $Pr(Y|X)$  and then classifies observations to the class with highest estimated probability.
- Given a positive integer  $K$  and a test observation  $X_{test}$ :
  - Identify  $K$  points in training data closest to  $X_{test}$ . Label  $\mathcal{N}_{test}$ .
  - Estimate conditional probability for class  $j$  as fraction of points in  $\mathcal{N}_{test}$  whose response values equal  $j$ :

$$Pr(Y = j|X = X_{test}) = \frac{1}{K} \sum_{i \in \mathcal{N}_{test}} \mathbb{I}(Y_i = j).$$

- Classify the test observation to class with the largest probability.

# Unsupervised Learning

Recall in unsupervised learning, we want to discover interesting patterns about the measurements  $X_1, X_2, \dots, X_p$  (often called features).

## Examples

- A cancer researcher might assay gene expression levels in patients with breast cancer and then look for subgroups among the samples or the genes, to obtain a better understanding of the disease.
- An online shopping site might try to identify groups of shoppers with similar purchase histories and items of particular interest to shoppers within each group. Then shoppers can be preferentially shown the items in which they are likely to be interested, based on the purchase histories of similar shoppers.
- A search engine might choose which search results to display to an individual based on the click histories of other individuals with similar search patterns.

# Supervised vs. Unsupervised Learning

## Supervised Learning

- As discussed, lots of tools at our disposal: regression, KNN, curve fitting, etc.
- All have a clear goal in mind: prediction!
- Also have a clear understanding of how to assess the quality: estimate test error with an independent test set or cross-validation.

# Supervised vs. Unsupervised Learning

## Supervised Learning

- As discussed, lots of tools at our disposal: regression, KNN, curve fitting, etc.
- All have a clear goal in mind: prediction!
- Also have a clear understanding of how to assess the quality: estimate test error with an independent test set or cross-validation.

## Unsupervised Learning

- Much more challenging and subjective – no clear goal like prediction.
- Often performed as part of *exploratory data analysis*.
- No real way to assess the quality of results; no 'true answer' to check work against.

## Types of Questions

- Is there an informative way to visualize the data?
- Can we discover subgroups among the variables or among the observations?

# Unsupervised Learning

## Types of Questions

- Is there an informative way to visualize the data?
- Can we discover subgroups among the variables or among the observations?

## Today

- **Principle Component Analysis:** A tool for data visualization or pre-processing before supervised techniques are applied.
- **Clustering:** Class of methods for discovering unknown subgroups in data.

# Principal Component Analysis (PCA)

# Principal Component Analysis (PCA) <sup>1</sup>

## What Are Principle Components?

When faced with a large set of correlated variables, or features, principle components allow us to summarize this set with a smaller number of representative variables that collectively explain most of the variability of the original set.

---

<sup>0</sup>“Some figures taken from “An Introduction to Statistical Learning” (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani”



# Principal Component Analysis (PCA) <sup>1</sup>

## What Are Principle Components?

When faced with a large set of correlated variables, or features, principle components allow us to summarize this set with a smaller number of representative variables that collectively explain most of the variability of the original set.

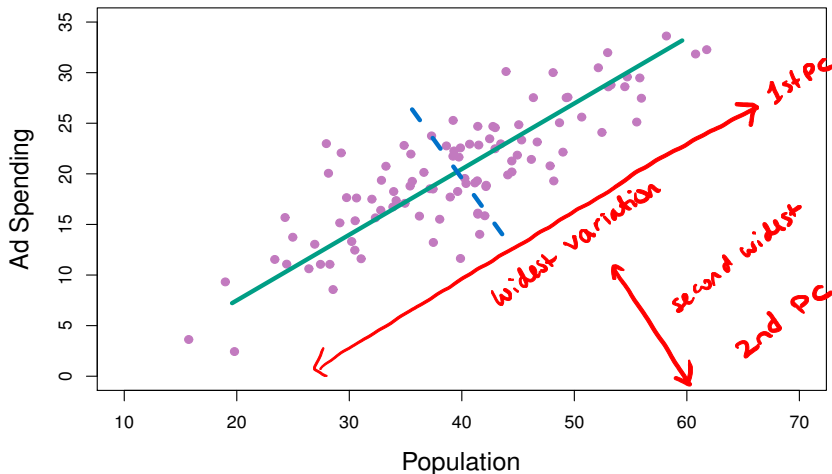
- Can think of this as a dimension reduction technique.
- The principle component directions are directions in the feature space along which the original data are *highly variable*.
- They also define subspaces *as close as possible* to the data.

---

<sup>0</sup>“Some figures taken from “An Introduction to Statistical Learning” (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani”

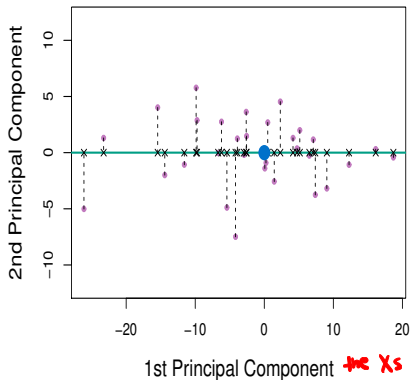
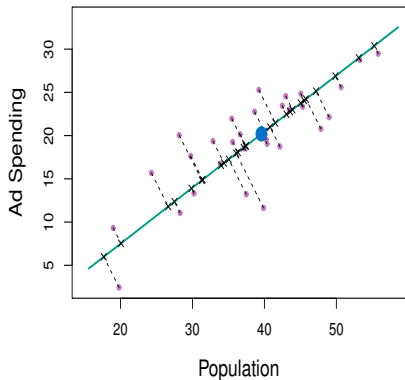
# Principal Component Analysis (PCA)

The **first principle component (PC)** direction of the data is that along which the observations *vary the most*.



# Principal Component Analysis (PCA)

Meaning if we *project* the observations onto the first PC line, the resulting projected observations have the largest possible variance. (Projecting onto any other line would yield projected values with lower variance.)



# Principal Component Analysis (PCA)

## Another Interpretation

The first PC defines a line *as close as possible* to the data.

# Principal Component Analysis (PCA)

## Another Interpretation

The first PC defines a line *as close as possible* to <sup>all of</sup> the data. <sup>points.</sup>

## Interpretation of the PC Score

- Recall the advertising budget is plotted against city populations for a company in 100 different cities.
- Think of the values of the first PC as single number summaries of the joint population and ad budgets for each location.
- A city with a negative first PC score suggest below-average population size and below-average ad spending.

## Finding the First PC Mathematically

- The first PC of a set of covariates  $X_1, X_2, \dots, X_p$  is the normalized linear combination of the predictors

$$\text{vector } \underline{Z}_1 = \phi_{11}\underline{X}_1 + \phi_{21}\underline{X}_2 + \dots + \phi_{p1}\underline{X}_p \rightarrow \text{calculating weights of } \phi$$

that has the largest variance.

- By normalized, we mean that  $\sum_{j=1}^p \phi_{j1}^2 = 1$ .  $\rightarrow$  all weights are = 1.
- We refer to the elements  $\phi_{11}, \dots, \phi_{p1}$  as the ~~loadings~~ <sup>weights</sup> of the first PC.
- Typically the predictors are centered:  $X_j \equiv X_j - \bar{X}_j$ ,  $j = 1, \dots, J$ .

Can see which features have the most contribution (biggest weight)

## Computing the First PC

- Boils down to an optimization procedure:

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

- Because the predictors are centered, the objective function is just the sample variance of the  $n$  values of  $z_{i1}$ , i.e. we are maximizing

$$\frac{1}{n} \sum_{i=1}^n z_{i1}^2.$$

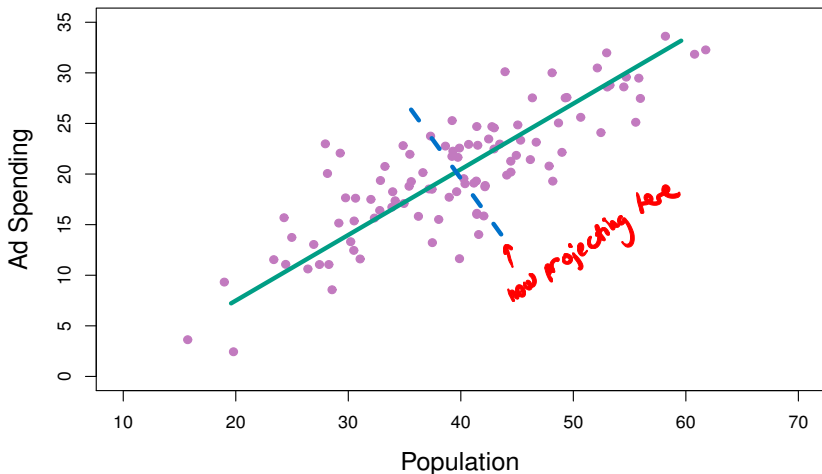
# The Second PC

- In general, can construct up to  $p$  PCs (well, really  $\min(n - 1, p)$  PCs).
- The second PC, denoted  $Z_2$ , is a linear combination of the variables that is *uncorrelated* with the first PC,  $Z_1$ , and has the largest variance subject to this constraint.



# Principal Component Analysis (PCA)

The zero correlation condition, is equivalent to saying the second PC must be *perpendicular* to the first PC.



## Computing the Second PC

- The second PC of a set of covariates  $X_1, X_2, \dots, X_p$  is the normalized linear combination of the predictors

$$Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \dots + \phi_{p2}X_p$$

that has the largest variance and is *uncorrelated* with  $Z_1$ .

- Computing the *second* PC involves the optimization procedure

$$\max_{\phi_{12}, \dots, \phi_{p2}} \left\{ \frac{1}{n} \sum_{i=1}^n \left( \sum_{j=1}^p \phi_{j2} x_{ij} \right)^2 \right\}$$

subject to

$$\sum_{j=1}^p \phi_{j2}^2 = 1 \quad \text{and} \quad \sum_{i=1}^n z_{i1} z_{i2} = 0.$$

# Principle Component *Analysis*

## PCA

PCA refers to the process of finding the principle components and then using these components to better understand the data.

# Principle Component *Analysis*

## PCA

PCA refers to the process of finding the principle components and then using these components to better understand the data.

- We could, for example, examine data  $X_1, X_2, \dots, X_p$  by looking at all two-dimensional scatterplots of the data, but for large  $p$ , there are lots of them.
- A better idea is to find a low-dimensional representation of the data that captures as much information as possible and view the data through this representation.
- PCA is a tool for this.

## *Practically Computing PCs*

- Computing the PCs boils down to an eigenvalue problem.
- The PC directions  $\phi_1, \phi_2, \dots, \phi_p$  are the ordered sequence of eigenvectors of the matrix  $X^T X$ .
- In R the function `prcomp()` runs a PCA analysis on a data set.

# Basic PCA Example

```
> # install.packages("ISLR")  
> library("ISLR")  
> head(USArrests)
```

	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

# Basic PCA Example

```
> USArrests <- apply(USArrests, 2, scale)
> head(USArrests)
```

	Murder	Assault	UrbanPop	Rape
[1,]	1.24256408	0.7828393	-0.5209066	-0.003416473
[2,]	0.50786248	1.1068225	-1.2117642	2.484202941
[3,]	0.07163341	1.4788032	0.9989801	1.042878388
[4,]	0.23234938	0.2308680	-1.0735927	-0.184916602
[5,]	0.27826823	1.2628144	1.7589234	2.067820292
[6,]	0.02571456	0.3988593	0.8608085	1.864967207

# Basic PCA Example

→ best pca function

```
> pca <- prcomp(USArrests)
```

```
> pca
```

Standard deviations:

[1] 1.5748783 0.9948694 0.5971291 0.4164494

Rotation:

	PC1	PC2	PC3	PC4
Murder	-0.5358995	0.4181809	-0.3412327	0.64922780
Assault	-0.5831836	0.1879856	-0.2681484	-0.74340748
UrbanPop	-0.2781909	-0.8728062	-0.3780158	0.13387773
Rape	-0.5434321	-0.1673186	0.8177779	0.08902432

✓  
if you square all the values and add them up they will equal 1

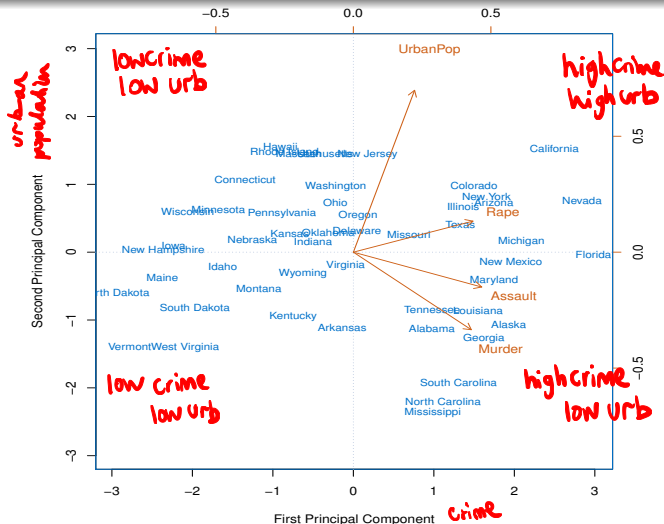


## Interpretation of the PCs

- First loading places approximately equal weight on Assault, Murder, and Rape, and much less weight on UrbanPop.
  - Think of it as a measure of overall rates of serious crimes.
- The second loading places most weight on UrbanPop and less weight on the others.
  - Think of it as a measure of the urbanization of a state
- Tells us that crime-related variables are correlated with each other, but urban population isn't as correlated with the other three.

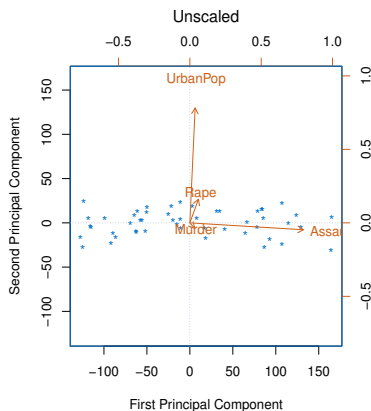
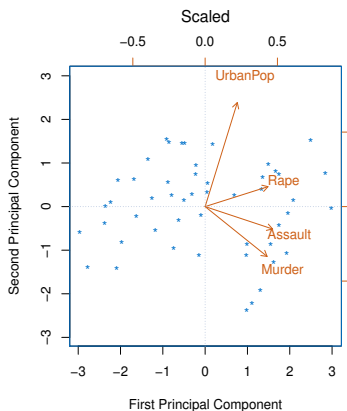
# Basic PCA Example

A plot of the first two PC scores and the loading vectors. You can reproduce a (less-pretty) version with `biplot(pca)`.



# Basic PCA Example

We mentioned that the variables should be *centered* to have mean 0, but usually want to *scale* variables as well to have  $sd = 1$ .



## The Next Question...

How much of the information in a given data set is lost by projecting the observations onto the first few PCs?

## The Next Question...

How much of the information in a given data set is lost by projecting the observations onto the first few PCs?

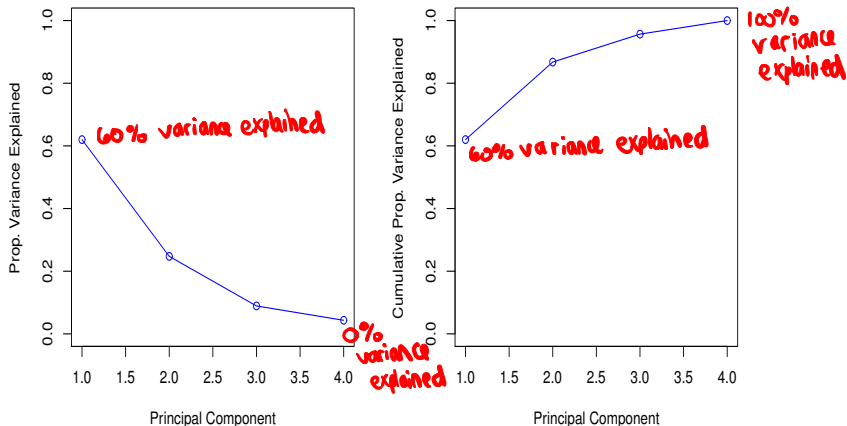
## Percent Variance Explained

- To answer this, we are interested in the proportion of variance explained (PVE) by each PC.
- The PVE of the  $m^{th}$  principal component is

$$\frac{\sum_{i=1}^n \left( \sum_{j=1}^p \phi_{jm} x_{ij} \right)^2}{\sum_{i=1}^p \sum_{i=1}^n x_{ij}^2} = \frac{\text{Variance explained by } m^{th} \text{ PC}}{\text{Total Variance}}$$

# Basic PCA Example

Plots of the individual PVE and cumulative PVE for the four PCs.



# K-Means Clustering

# Clustering

- **Clustering** refers to a broad set of techniques for finding subgroups, or clusters, in a dataset.
- The idea is to partition the observations into distinct groups so that observations within each group are similar, while observations in different groups are different from each other.



- **Clustering** refers to a broad set of techniques for finding subgroups, or clusters, in a dataset.
- The idea is to partition the observations into distinct groups so that observations within each group are similar, while observations in different groups are different from each other.

## Similarities to PCA

Both clustering and PCA seek to simplify the data via a small number of summaries:

- PCA looks to find a low-dimensional representation of the observations that explain most of the variance.
- Clustering looks to find homogenous subgroups among the observations.

# K-Means Clustering

## K-Means Clustering

Simple approach for partitioning a data set into  $K$  distinct, non-overlapping clusters. Note  $K$  is pre-specified.

# K-Means Clustering

## K-Means Clustering

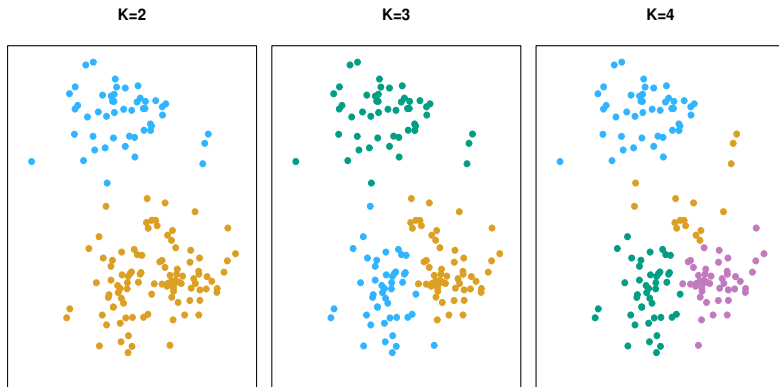
Simple approach for partitioning a data set into  $K$  distinct, non-overlapping clusters. Note  $K$  is pre-specified.

## How Do We Do It?

- First we specify the number of desired clusters  $K$ .
- The  $K$ -means algorithm then assigns each observation to exactly one of the  $K$  clusters.
- The algorithm boils down to a simple and intuitive mathematical problem.

# Example of $K$ -means<sup>2</sup>

A simulated dataset with 150 observations in two-dimensional space. We see the results of the  $K$ -mean algorithm using different values of  $K$ .



<sup>1</sup>“Some figures taken from “An Introduction to Statistical Learning” (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani”

## Notation

Let  $C_1, C_2, \dots, C_K$  denote sets containing the indices of the observations in each cluster. These sets satisfy the following properties:

1.  $C_1 \cup C_2 \cup \dots \cup C_K = \{1, \dots, n\}$ . (Each observation belongs to at least one of the  $K$  clusters.)
2.  $C_k \cap C_{k'} = \emptyset$ . (The clusters are non-overlapping.)

# K-Means Clustering

## Main Idea

The idea behind  $K$ -means clustering is that a *good* clustering is one for which the *within-cluster variation* is small as possible.

# K-Means Clustering

## Main Idea

The idea behind  $K$ -means clustering is that a *good* clustering is one for which the *within-cluster variation* is small as possible.

## Within-Cluster Variation

- For cluster  $C_k$ , denote the *within-cluster variation* by  $W(C_k)$ .
- $W(C_k)$  measures the amount by which the observations within a cluster differ within each other.
- The algorithm is then an optimization problem:

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

smallest distance  
from center  
of cluster



# K-Means Clustering

## Optimization Task

- To solve the optimization problem, we need to define  $W(C_k)$ .
- The most common choice is to use *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.

- In words this is the sum of the pairwise squared Euclidean distances between observations in the  $k^{th}$  cluster, divided by the total number of observations in the  $k^{th}$  cluster.



# K-Means Clustering

## Optimization Task

- To solve the optimization problem, we need to define  $W(C_k)$ .
- The most common choice is to use *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.

- In words this is the sum of the pairwise squared Euclidean distances between observations in the  $k^{th}$  cluster, divided by the total number of observations in the  $k^{th}$  cluster.
- Consequently, we want to solve the optimization problem

*locally this  
solves the  
problem.*

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

# K-Means Clustering

solving this problem globally is hard!

## How do we Solve this Problem?

- Want to come up with an algorithm that partitions the data such that the following is minimized:

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

- This is a hard problem – there are around  $K^n$  ways to partition  $n$  observations in to  $K$  clusters!
- The K-means clustering algorithm is very simple and can be shown to provide a *pretty good* solution.

↳ local  
not guaranteed to be the best.

# K-Means Clustering


## K-Means Clustering Algorithm

1. Randomly assign a number, from 1 to  $K$ , to each observation. This serves as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing: → center of values
  - a. For each of the  $K$  clusters, compute the cluster *centroid*. The  $k^{th}$  centroid is the vector of the  $p$  feature means (covariate 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).

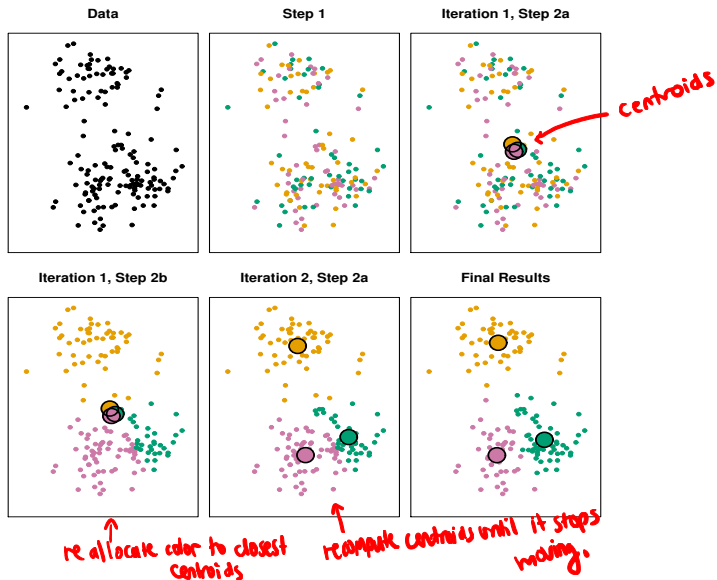
# K-Means Clustering

## K-Means Clustering Algorithm

1. Randomly assign a number, from 1 to  $K$ , to each observation. This serves 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  $k^{th}$  centroid is the vector of the  $p$  feature means (covariate 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).

- This algorithm provides a **local minimum** not global. 
- We must decide on how many clusters  $K$  exist in the data beforehand.

# K-Means Clustering



# K-Means Clustering

## Why does the Algorithm Work?

- Notice the following identity

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

↗ always get closer to local min no matter what

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

- Reallocating the observations (step 2b) can only improve the above, thereby always decreasing the value of the objective function in the optimization problem.
- As the algorithm runs, the clustering obtained will continually improve until the result no longer changes.

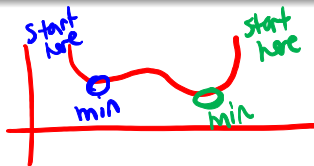
## The Initial Cluster Matters

- Recall, the algorithm finds a local rather than a global optimum.

# K-Means Clustering

## The Initial Cluster Matters

- Recall, the algorithm finds a local rather than a global optimum.
- Therefore the results will depend on the initial (random) cluster assignment of each observation in step 1.
- It's important to run the algorithm multiple times from different random initial clusterings. Then select the *best* solution as determined by the objective function.



so run multiple times!



# K-Means Clustering

*K*-means performed six times with different starting assignments gets six different values of the objective.



# Example of K-means

Simulate data in which there are truly two clusters.

```
> set.seed(2)
> # Create a random matrix where the first column is
> # N(3, 1) and the 2nd column N(-4, 1)
>
> x <- matrix(rnorm(50*2), ncol = 2)
> x[1:25, 1] <- x[1:25, 1] + 3
> x[1:25, 2] <- x[1:25, 2] - 4
> head(x, 5)
```

```
      [,1]      [,2]
[1,] 2.103085 -4.838287
[2,] 3.184849 -1.933699
[3,] 4.587845 -4.562247
[4,] 1.869624 -2.724284
[5,] 2.919748 -5.047573
```

# Example of K-means

Run the  $K$ -means algorithm with  $K = 2$ .

```
> km.out <- kmeans(x, centers = 2, nstart = 20)
> # Cluster assignments for the 50 observations
>
> km.out$cluster[1:25]
```

```
[1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

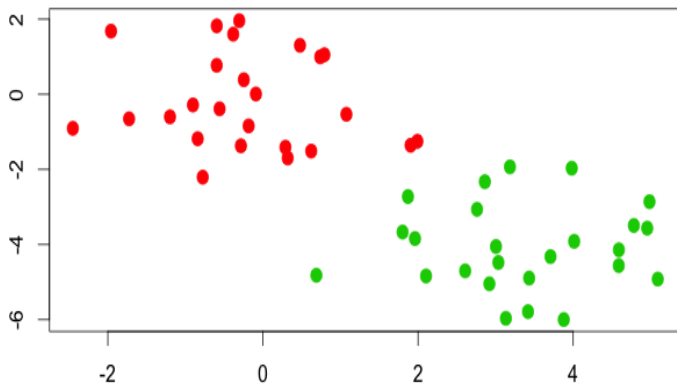
```
> km.out$cluster[26:50]
```

```
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
> plot(x, col = (km.out$cluster + 1),
+      main = "K-Means Clustering Results with K=2",
+      xlab = "", ylab = "", pch = 20, cex = 2)
```

# Example of K-means

**K-Means Clustering Results with K=2**



# Example of K-means

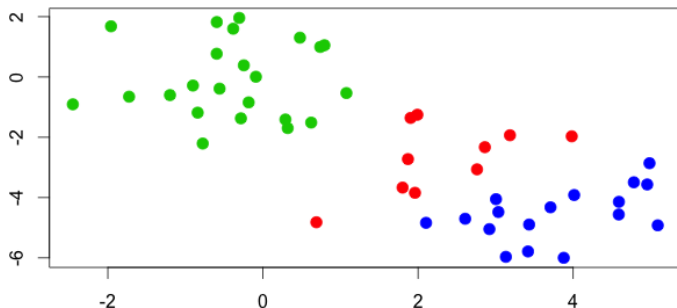
Run the  $K$ -means algorithm with  $K = 3$ .

```
> set.seed(4)
> km.out <- kmeans(x, centers = 3, nstart = 20)
> km.out$cluster
```

```
[1] 3 1 3 1 3 3 3 1 3 1 3 1 3 1 3 1 3 3 3 3 3 1 3 3 3 2 2 2
[29] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2
```

# Example of K-means

**K-Means Clustering Results with K=3**



# Example of K-means

## Initial Cluster Matters

- To run the `kmeans()` function in R with multiple initial cluster assignments, we use the `nstart` argument.
- If a value of `nstart` greater than one is used, then K-means clustering will be performed using multiple random assignments in Step 1 of the *K*-means algorithm.
- `km.out$tot.withinss` is the total within-cluster sum of squares. This is what we seek to minimize.

# Example of K-means

```
> set.seed(3)
> km.out <- kmeans(x, centers = 3, nstart = 1)
> km.out$tot.withinss
```

```
[1] 104.3319
```

```
> km.out <- kmeans(x, centers = 3, nstart = 20)
> km.out$tot.withinss
```

```
[1] 97.97927
```



- Chapter 10 (Unsupervised Learning) in An Introduction to Statistical Learning.
- Hadley Wickham, “Reshaping Data with the reshape Package”, *Journal of Statistical Software* 21 (2007): 12, <http://www.jstatsoft.org/v21/i12>.
- Hadley Wickham, “The Split-Apply-Combine Strategy for Data Analysis”, *Journal of Statistical Software* 40 (2011): 1, <http://www.jstatsoft.org/v40/i1>