DATASCI 347 Machine Learning

Lecture 16: PCA, matrix completion, clustering

Ruoxuan Xiong

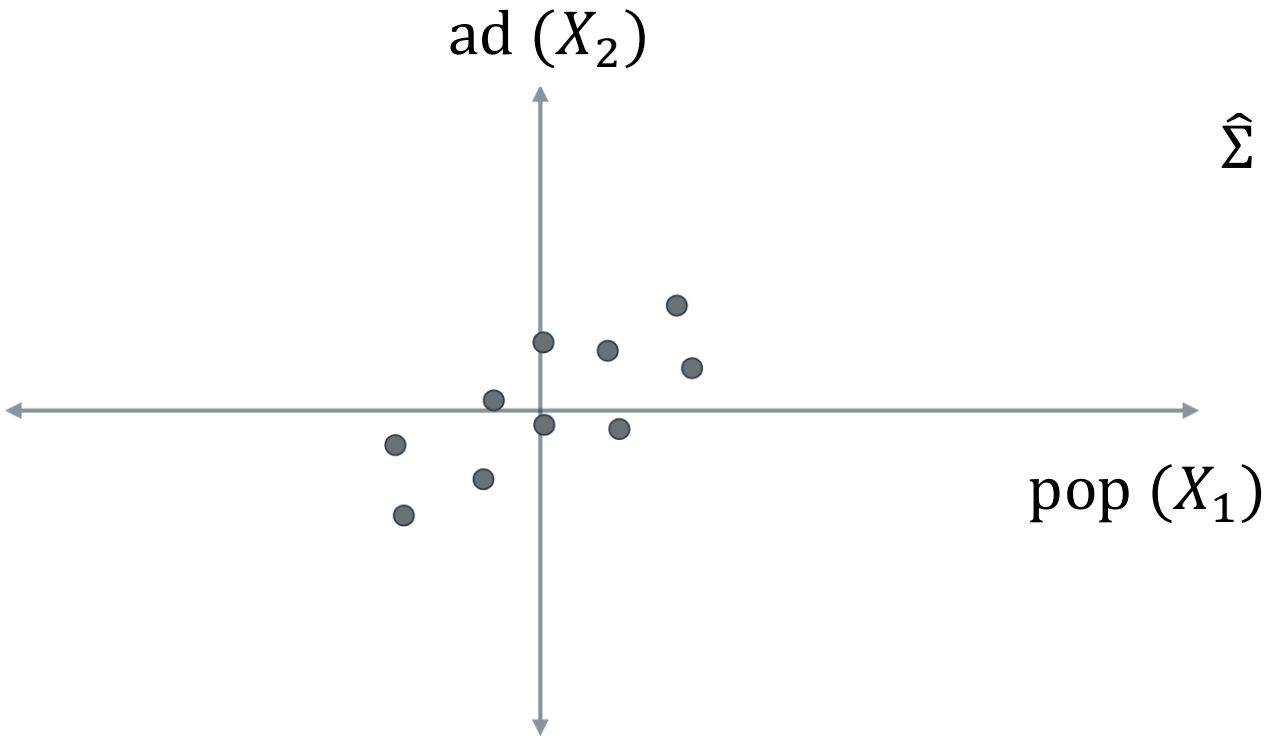
Suggested reading: ISL Chapter 6 and 12

Lecture plan

- PCA
- Matrix completion
- Clustering methods

How to perform PCA I

1. Estimate the **covariance matrix** $\hat{\Sigma}$ of $X_1, X_2, \cdots, X_p$.
   - $\hat{\Sigma}$ is a $p \times p$ matrix, the (i, j)-th entry being the covariance of X_i, X_j.
   - **Example**: population size (pop) and ad spending (ad) for 100 cities.



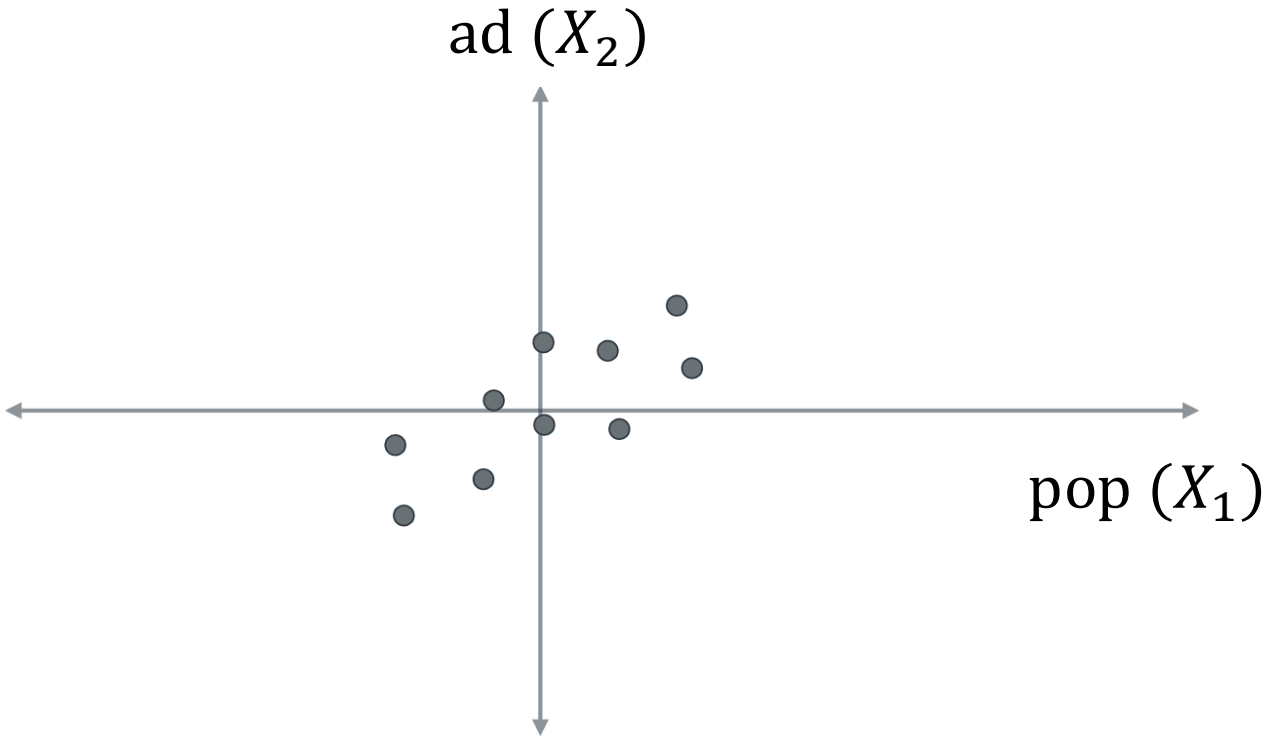
$$\hat{\Sigma} = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) \\ \mathrm{Cov}(X_1, X_2) & \mathrm{Var}(X_2) \end{bmatrix}$$

$$\hat{\Sigma} = \begin{bmatrix} 3.816 & 1.826 \\ 1.826 & 2.184 \end{bmatrix}$$

How to perform PCA II

2. Calculate the **eigenvalues** and **eigenvectors** of the covariance.

- **Covariance matrix**: $\hat{\Sigma} = \begin{bmatrix} 3.816 & 1.826 \\ 1.826 & 2.184 \end{bmatrix}$.



Unit norm eigenvectors

$\begin{pmatrix} 0.839 \\ 0.544 \end{pmatrix}$ $\begin{pmatrix} 0.544 \\ -0.839 \end{pmatrix}$

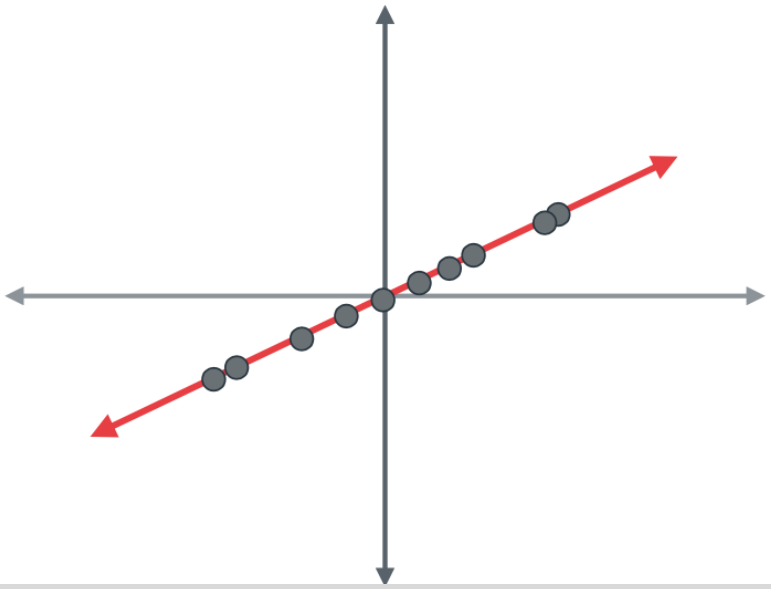
Eigenvalues

$\lambda_1 = 5$ $\lambda_2 = 1$

Projection to first principal component

3. Select the **first principal component**

- First principal component, which is corresponds to the following equation:
  - $z_{i1} = 0.839 \times (\text{pop}_i - \overline{\text{pop}}) + 0.544 \times \left(\text{ad}_i - \overline{\text{ad}}\right)$ and $Var(z_{i1}) = \lambda_1$



Unit norm eigenvectors (direction)

$$\begin{pmatrix} 0.839 \\ 0.544 \end{pmatrix}$$

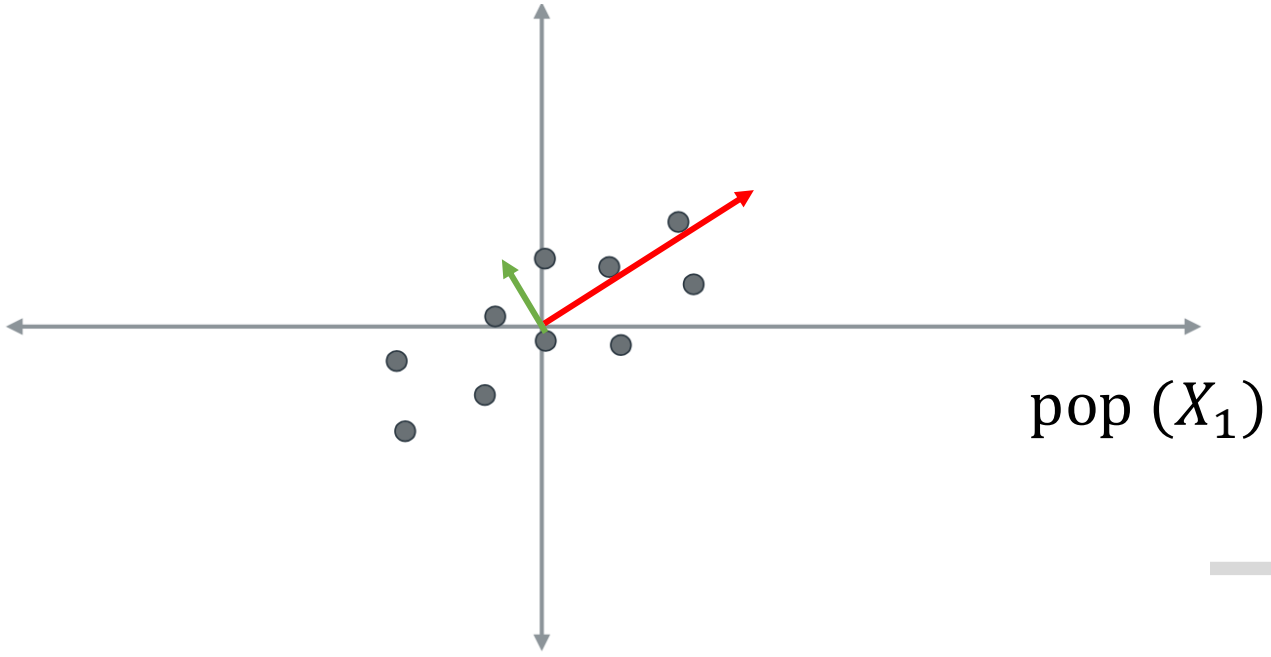
Eigenvalues (magnitude)

$\lambda_1 = 5$

How to perform PCA IV

4. Select the **second principal component** (if necessary)

- The second principal component Z_2 has **largest variance** subject to **being orthogonal** to first principal component Z_1
  - $z_{i2} = 0.544 \times (\text{pop}_i - \overline{\text{pop}}) - 0.839 \times (\text{ad}_i - \overline{\text{ad}})$ and $Var(z_{i2}) = \lambda_2$



Unit norm eigenvectors (direction)

$$\begin{pmatrix} 0.839 \\ 0.544 \end{pmatrix} \quad \begin{pmatrix} \mathbf{0.544} \\ \mathbf{-0.839} \end{pmatrix}$$

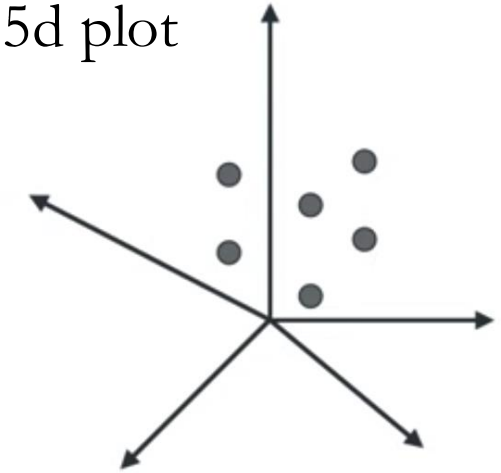
Eigenvalues (magnitude)

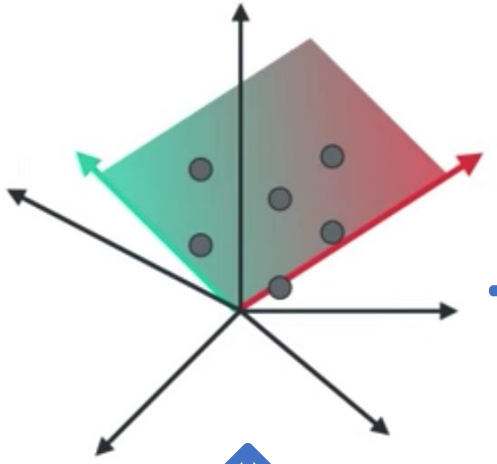
$\lambda_1 = 5$ ✓ $\lambda_2 = 1$ ✓

Summarizing PCA

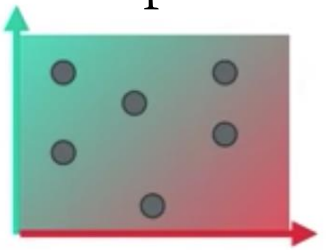
X_1	X_2	X_3	X_4	X_5
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*

5d plot





2d plot



Small table

Z_1	Z_2
*	*
*	*
*	*
*	*
*	*
*	*
*	*
*	*
*	*
*	*
*	*
*	*

Covariance matrix

*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*

Eigenvector Eigenvalue

V_1	λ_1
V_2	λ_2
V_3	λ_3
V_4	λ_4
V_5	λ_5

Big

Small

More on PCA

- **Mean**: Variables should be centered to have mean zero
  - First principal component (PC) reflects the direction of max variance, instead of the mean of the data
- **Variance**: Choose case by case whether to scale variables to have unit variance
  - Results typically *depend on* whether variables have been individually scaled
    - Small-scale variables will have small variance
  - *Whether to scale* depends on whether variables are *measured on the same unit*

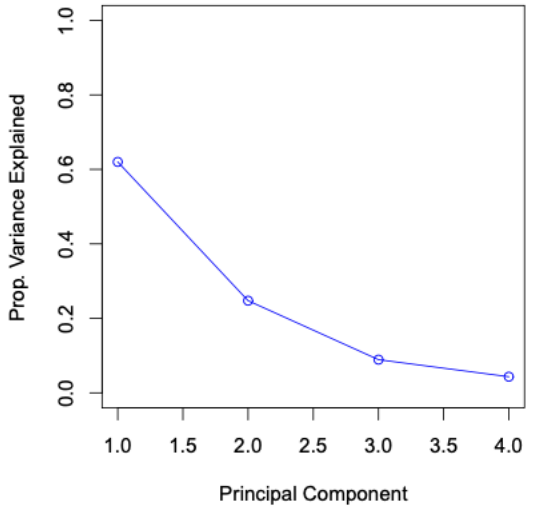
- Example 1: Variables are expression levels of genes (no need to scale the genes)

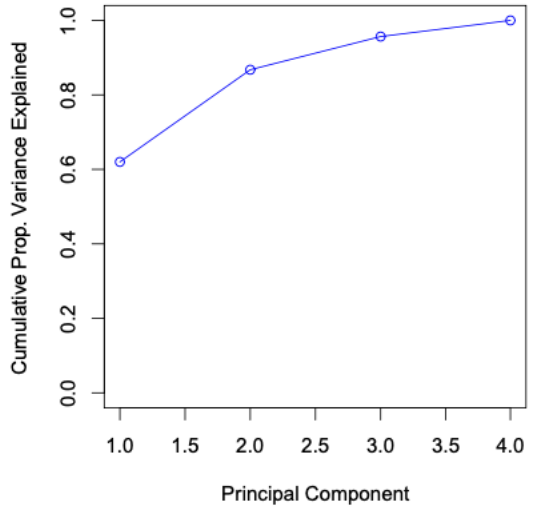
- Example 2: Variables include ad spending and population size (scale the variables)

Choosing the number of PCs

- **Choosing the number of PCs**:
  - How much information is lost by projecting observations on the first M PCs?
  - Equivalently, how much variance of the data is not contained in the first M PCs?
  - Choose the smallest number that explains a sizable amount of the variation
- Eigenvalues of feature covariance matrix: $\lambda_1, \lambda_2, \cdots, \lambda_p$
- **Scree plot** shows the variance explained by each PC (an ad hoc method):

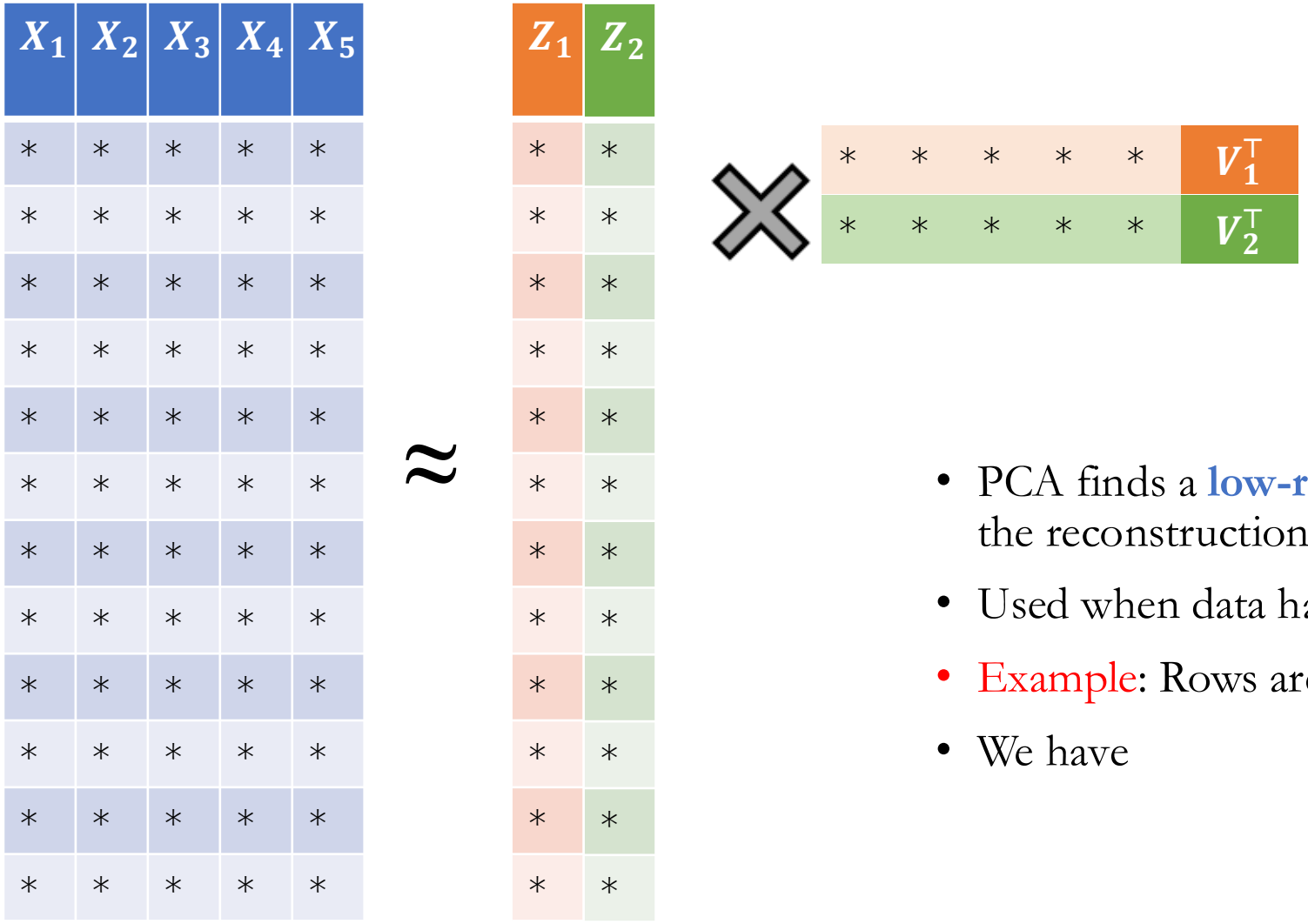
$$\frac{\lambda_1}{\lambda_1+\lambda_2+\cdots+\lambda_p}, \frac{\lambda_2}{\lambda_1+\lambda_2+\cdots+\lambda_p}, \cdots, \frac{\lambda_p}{\lambda_1+\lambda_2+\cdots+\lambda_p}$$





The first PC explains 62%
The next PC explains 24.7%

PCA for low-rank matrix factorization



- PCA finds a **low-rank matrix factorization** that minimizes the reconstruction error
- Used when data has **inherent low-dimensional structure**
- Example: Rows are users and columns are movies
- We have

$$\boldsymbol{X} \approx [\boldsymbol{Z_1} \quad \boldsymbol{Z_2}] \begin{bmatrix} \boldsymbol{V_1^\top} \\ \boldsymbol{V_2^\top} \end{bmatrix}$$

Missing values and matrix completion

- In data streaming services (e.g., Netflix, Amazon), most of the rating matrix is missing --- users only rated a tiny fraction of all movies/items
- We use the approximation

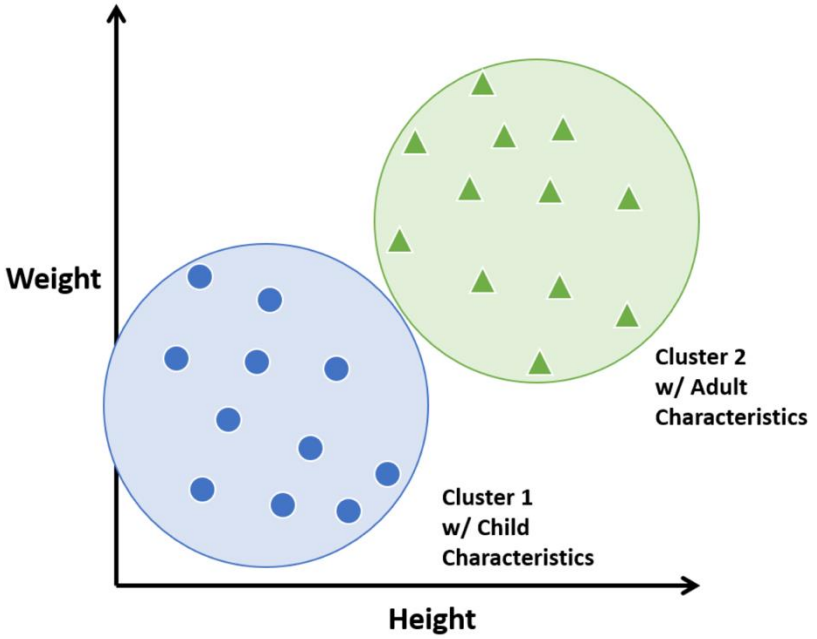
$$\boldsymbol{X} \approx [\boldsymbol{Z_1} \quad \boldsymbol{Z_2}] \begin{bmatrix} \boldsymbol{V_1^\top} \\ \boldsymbol{V_2^\top} \end{bmatrix}$$

  - Most entries in $\boldsymbol{X}$ are *missing*
  - $[\boldsymbol{Z_1} \quad \boldsymbol{Z_2}]$: *latent user features (e.g., cliques)*
  - $\begin{bmatrix} \boldsymbol{V_1^\top} \\ \boldsymbol{V_2^\top} \end{bmatrix}$: *latent movie features (e.g., genres)*
  - Estimate $\boldsymbol{Z}$ and $\boldsymbol{V}$ using observed entries in $\boldsymbol{X}$
  - An *iterative* algorithm:
    1. Impute missing entries by $\bar{X}$ (mean)
    2. Apply PCA or similar methods to estimate $\boldsymbol{Z}$ and $\boldsymbol{V}$
    3. Use estimated $\boldsymbol{Z}$ and $\boldsymbol{V}$ to impute missing entries in $\boldsymbol{X}$
    4. Repeat Steps 2 and 3 until convergence

	Jerry Maguire	Oceans	Road to Perdition	A Fortunate Man	Catch Me If You Can	Driving Miss Daisy	The Two Popes	The Laundromat	Code 8	The Social Network	⋯
Customer 1	•	•	•	•	4	•	•	•	•	•	⋯
Customer 2	•	•	3	•	•	•	3	•	•	3	⋯
Customer 3	•	2	•	4	•	•	•	•	2	•	⋯
Customer 4	3	•	•	•	•	•	•	•	•	•	⋯
Customer 5	5	1	•	•	4	•	•	•	•	•	⋯
Customer 6	•	•	•	•	•	2	4	•	•	•	⋯
Customer 7	•	•	5	•	•	•	•	3	•	•	⋯
Customer 8	•	•	•	•	•	•	•	•	•	•	⋯
Customer 9	3	•	•	•	5	•	•	1	•	•	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

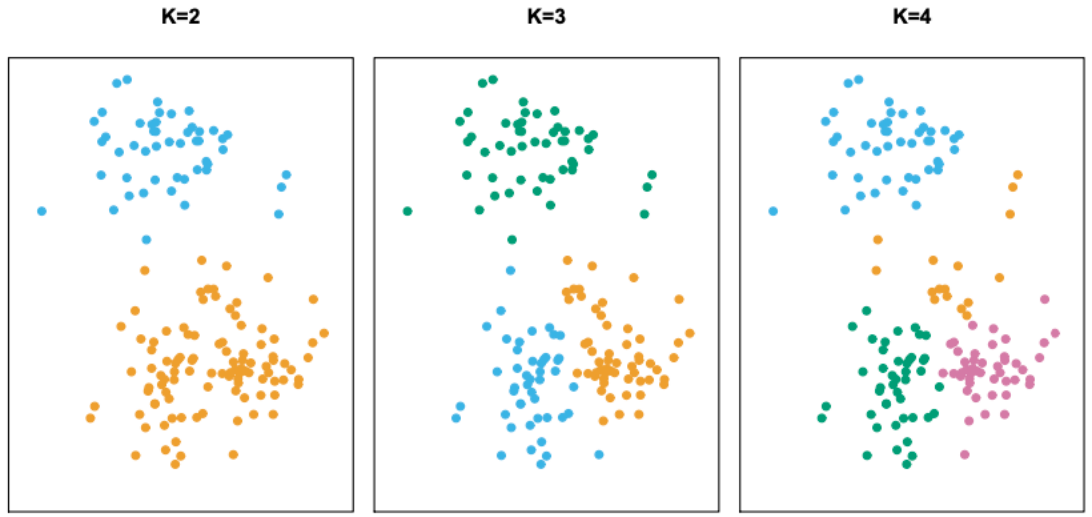
Clustering methods

- PCA looks to find a low-dimensional representation of observations
- Clustering looks to find homogeneous subgroups among observations
- A marketing example
  - Perform market segmentation by identifying subgroups who are more likely to purchase a particular product
  - Use features, e.g., median household income, occupation, distance form nearest urban area



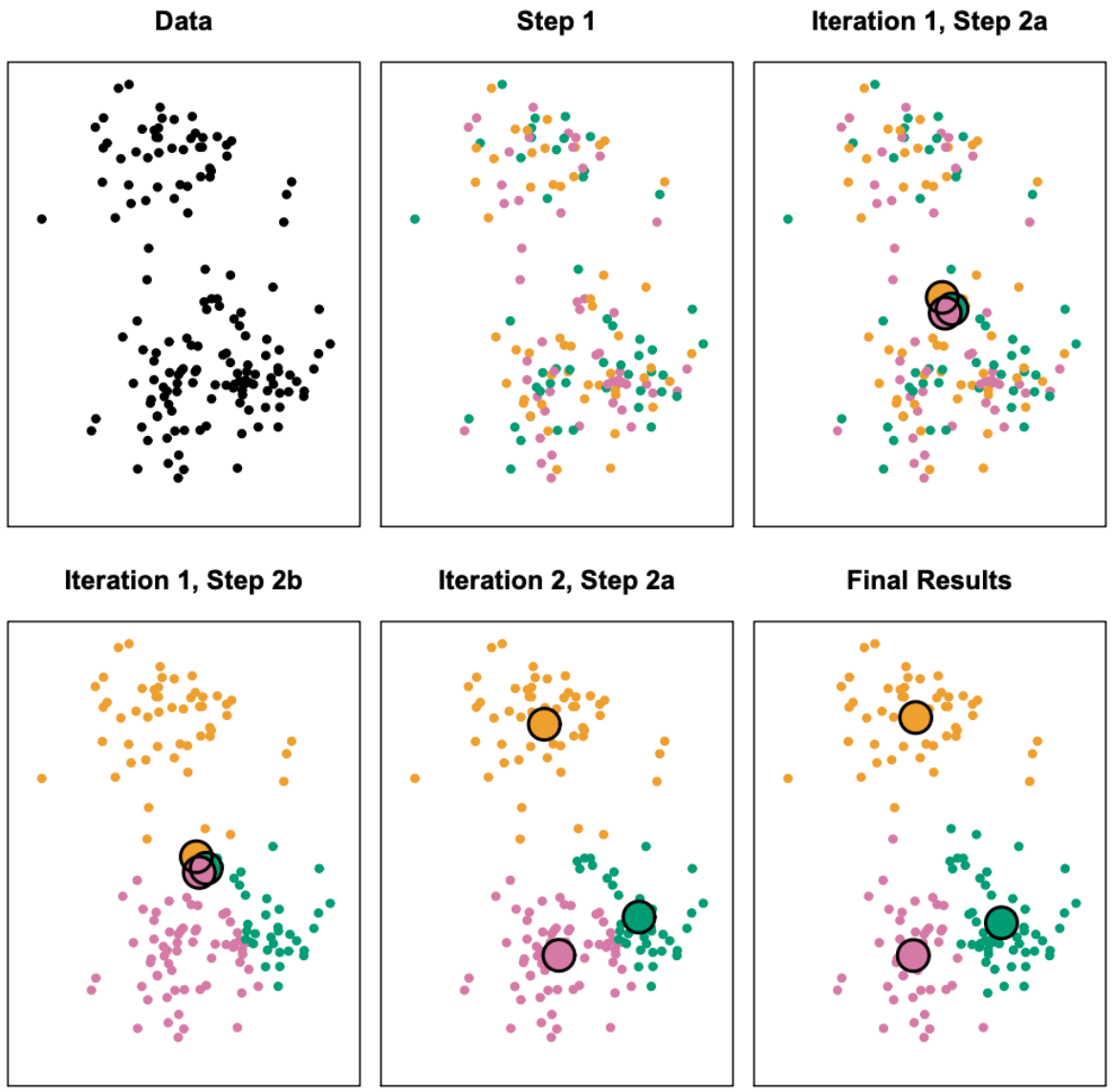
K-means clustering

- A *good clustering* is one for which the within-cluster variation is as small as possible
- **K-means clustering** is a simple and elegant approach for partitioning a data set into *K distinct, non-overlapping clusters*
  - Let $C_1, \cdots, C_K$ denote set of indices of observations in each cluster
    - $C_1 \cup \cdots \cup C_K = \{1, \cdots, n\}$: each observation belongs to at least one cluster
    - $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$: no observation belongs to more than one cluster



K-means clustering procedure

- *Randomly assign* a number, from 1 to K, to each observation, i.e., initial cluster assignment
- *Iterate* until the cluster assignments stop changing
  - For each cluster, *compute the cluster centroid*, i.e., vector of the p feature means for observations in the kth cluster
  - *Assign each observation* to the *cluster* whose *centroid is closest*



EMORY

Initialization of K-means clustering

- The *results* of K-means clustering will *depend on the initial cluster assignment*
- Solution: *run* the algorithm *multiple times* from different random initial configurations. Then one *selects the best solution*

K-means clustering performed six times with $K = 3$ and each time with a different initial cluster. Those labeled in red are optimal, with an objective value of 235.8.