STA 141A – Fundamentals of Statistical Data Science

Department of Statistics; University of California, Davis

Instructor: Dr. Akira Horiguchi (ahoriguchi@ucdavis.edu)
A01 TA: Zhentao Li (ztlli@ucdavis.edu)
A02 TA: Zijie Tian (zijtian@ucdavis.edu)
A03 TA: Lingyou Pang (lyopang@ucdavis.edu)

Section 9: Unsupervised learning

Spring 2025 (Mar 31 – Jun 05), MWF, 01:10 PM – 02:00 PM, Young 198

Comments

Supervised data: predictors $X_1, \ldots, X_p$ and a response Y measured on n observations.

Unsupervised data: predictors $X_1, \ldots, X_p$ measured on n observations, but no response.

- Still useful to analyze the association between the predictors $X_1, \ldots, X_p$.
- Often performed as part of an exploratory data analysis.
- Harder to assess the results from an unsupervised learning method; there is no "truth" to compare to.
  (In contrast, in supervised learning the "truth" is the response Y.)

Common unsupervised learning tasks: *clustering* and *dimension reduction*.

Overview

Based on Chapter 12 of ISL book James et al. (2021).

- For more R code examples, see R Markdown files in `https://www.statlearning.com/resources-second-edition`

Section 9: Unsupervised learning

- *K*-means clustering
- Principal component analysis (PCA)

Unsupervised learning

K-means clustering

Task: find homogeneous subgroups (i.e., *clusters*) among observations.

- "Market segmentation" aims to identify subgroups of people who might be more receptive to certain kind of advertisements/products etc. (TikTok)
- Flow cytometry: group cells based on their biomarker values.

Figure 1: From James et al. (2021).

If we index the n observations by the integers $1, 2, 3, \ldots, n$, then

$$\text{cluster } n \text{ observations} \iff \text{cluster the integers } 1, 2, 3, \ldots, n$$

In other words, we want to partition the set $\{1, 2, 3, \ldots, n\}$.

Definition (Cluster)

Clusters are sets $C_1, \ldots, C_K$ with the following features:

- $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, n\}$ (each obsn belongs to at least one cluster);
- $C_k \cap C_l = \emptyset$ for all $k \neq l$ (no observation belongs to more than one cluster).

How to select “best” clustering of given data?

- Two common techniques: *K-means clustering* and *hierarchical clustering*

K-means clustering partitions observations into K non-overlapping clusters.

- The user chooses the value of K before performing K-means clustering.
- "Good" clustering: if the obsns in each cluster are close to each other, i.e., if the *within-cluster variation* is relatively small.
- Several ways to define *within-cluster variation*. The most common choice involves the squared Euclidian distance (common distance of vectors).
- For obsns $x_1, \ldots, x_n \in \mathbb{R}$, within-cluster variation of a cluster C defined by

$$W(C) := \frac{1}{\#C} \sum_{i,i' \in C} (x_i - x_{i'})^2 . \tag{1}$$

- For obsns $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, within-cluster variation of a cluster C defined by

$$W(C) := \frac{1}{\#C} \sum_{i,i' \in C} \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2^2 = \frac{1}{\#C} \sum_{i,i' \in C} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2 . \tag{2}$$

- We want to find clusters $C_1, \ldots, C_K$ that minimize $\sum_{k=1}^{K} W(C_k)$.

Goal: We want to find clusters $C_1, \ldots, C_K$ that minimize $\sum_{k=1}^{K} W(C_k)$.

- This minimization problem is very difficult to solve precisely, since there are almost K^n ways to partition n observations into K clusters.
- The following algorithm can be shown to provide a local optimum.

K-means algorithm:

1. Randomly assign a number from 1 to K (K is pre-defined) to each obsn.
2. Iterate steps (a) and (b) until the cluster assignments stop changing:
   (a) For each cluster, compute *cluster centroid* (mean of all obsns in the cluster).
   (b) Assign each observation to the cluster whose centroid is the closest.

Example: draw & compute the centroid of the cluster $\{(1,2),(2,1),(3,2),(1,0)\}$

Comments:

- K-means clustering derives its name from the fact that the cluster centroids are computed as the mean of each cluster's observations.
- Step 2 can be shown to never increase $\sum_{k=1}^{K} W(C_k)$ — will reduce it until at local minimum. Value of obtained local minimum will depend on initial (random) cluster assignment in Step 1.
- To reduce prob. of choosing a "bad" local minimum, one should run the algorithm many times, and then choose clustering w/smallest $\sum_{k=1}^{K} W(C_k)$.

Simulation of K-means clustering

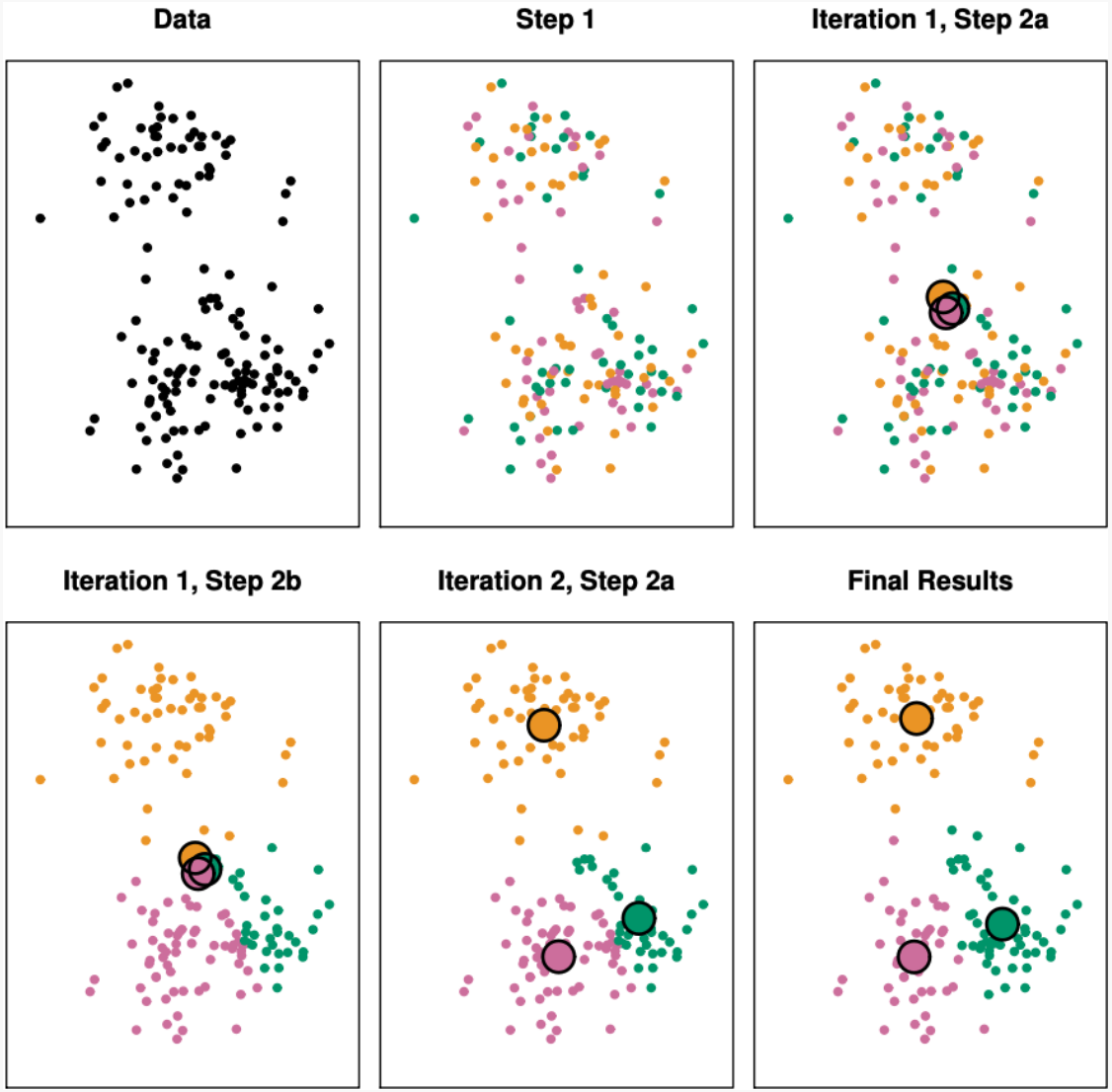

Figure 2: From James et al. (2021). 3-means clustering and 10 iterations.

Issues in clustering

- Should observations first be standardized in some way? E.g. should variables be scaled to have standard deviation one?
- Hierarchical clustering:
  - What dissimilarity measure should be used?
  - What type of linkage should be used?
  - Where shall the dendogram be cut (i.e., how many clusters do we need/want)?
- K-means clustering: how many clusters should we look for?

It is challenging to validate obtained clusters

- Outside scope of class; more details found in "sequel" book *The Elements of Statistical Learning*
- In practice, try several different choices, and look for the one with the most useful or interpretable solution.

Unsupervised learning

Principal component analysis (PCA)

Dimension reduction

Suppose we have n obsns on a set of p features $X_1, X_2, \ldots, X_p$.

- That is, suppose we have n data points $(x_{11}, x_{12}, \ldots, x_{1p})^\top, (x_{21}, x_{22}, \ldots, x_{2p})^\top, \ldots, (x_{n1}, x_{n2}, \ldots, x_{np})^\top$.
- Dimension reduction: reduce dimensionality of data while retaining as much information about the data as possible.
- Idea: not all p dimensions are equally interesting.
  E.g., if jth feature has almost the same value for all n obsns,
  do we really need to keep track of jth feature?
- Dimension reduction is useful for e.g.,
  - 2-dim scatterplots of data if $p = 10$.
  - image compression.
  - denoising images.

PCA – Idea

Principal components analysis (PCA) seeks a small number of dimensions that are as interesting as possible.

- Here "interesting" is measured by how much the n observations vary along the dimension.
- Each dimension (i.e., *principal component*) found by PCA is a *linear combination* of the p features.
  - A linear combination of the p features is defined as

    $$\phi_1 X_1 + \phi_2 X_2 + \cdots + \phi_p X_p$$

    for some *coefficient* values $\phi_1, \phi_2, \ldots, \phi_p$.
  - Given $X_1, X_2, \ldots, X_p$, a linear combination can be represented by the vector $\phi = (\phi_1, \phi_2, \ldots, \phi_p)$.
  - A linear combination is called *normalized* if $\|\phi\|_2^2 = 1$. (unit vector)
- Coefficients $\phi_1, \phi_2, \ldots, \phi_p$ of a principal component are called its *loadings*; the vector ϕ is called the principal component's *loading vector*.
- Principal component loading vectors are always orthogonal to each other.
  - Vectors $(a_1, \ldots, a_p)$ and $(b_1, \ldots, b_p)$ are called *orthogonal* or *perpendicular* to each other if $\sum_{j=1}^{p} a_j b_j = 0$ (i.e., if dot product is zero).
- PCA computes p orthogonal principal component loading vectors in order from "most interesting" to "least interesting".

First principal component

Given an $n \times p$ data matrix $\mathbf{X}$, how to compute the *first principal component*?

- Because we are only interested in variance, assume each column in $\mathbf{X}$ has mean zero.
- Coefficients $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ lead to the n values

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \cdots \phi_{p1} x_{ip} = \sum_{j=1}^{p} \phi_{j1} x_{ij} \qquad i = 1, \ldots, n. \tag{3}$$

- The first principal component loading vector is defined to be the **normalized** vector $(\phi_{11}, \phi_{21}, \ldots, \phi_{p1})$ that maximizes

$$\frac{1}{n} \sum_{i=1}^{n} z_{i1}^2 \,. \tag{4}$$

- (Normalization ensures the variance is not arbitrarily large.)
- Because each column of $\mathbf{X}$ is centered, mean of $z_{11}, z_{21}, \ldots, z_{n1}$ is also zero. So Eq. (4) is (almost) just the sample variance of $z_{11}, z_{21}, \ldots, z_{n1}$.
- (4) can be maximized via *eigen decomposition* or *singular value decomposition* (see lab, but don't need to know this for exam).

First principal component

- Geometric interpretation: the first principal component loading vector defines a direction in feature space along which the data vary the most.
- For all $i = 1, \ldots, n$, the data point x_i projected onto this direction gives the principal component score z_{i1}.

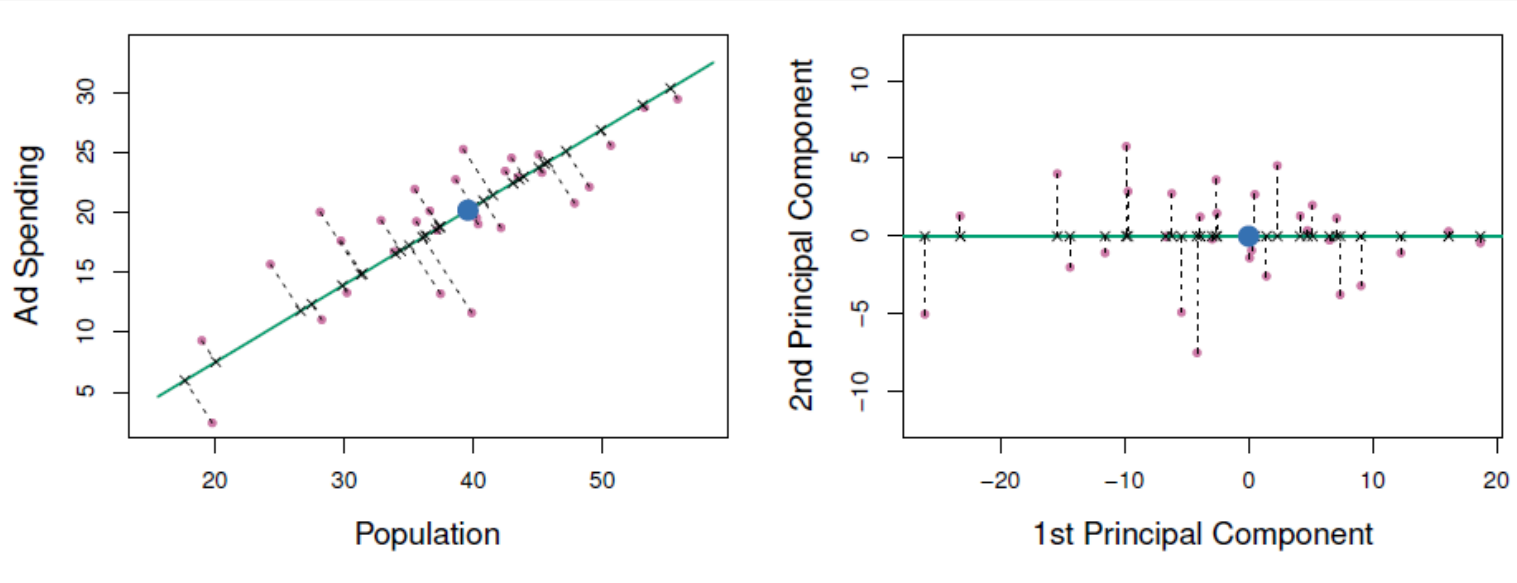

Figure 3: Figure by James et al. (2021). A subset of the advertising data. The mean pop and ad budgets are indicated with a blue circle. Left: The 1st principal component direction (in green). It is the dimension along which the data vary the most, and it defines the line that is closest to all n observations. Right: The left-hand panel has been rotated so that the 1st principal component direction coincides with the x-axis.

Remaining principal components

Suppose we have computed the first principal component loading vector ϕ_1.
How to compute the remaining principal components vectors?

PCA computes the second principal component by finding the normalized vector ϕ_2 that maximizes the sample variance $\frac{1}{n}\sum_{i=1}^{n} z_{i2}^2$, but now this vector ϕ_2 must be orthogonal to the vector ϕ_1.

- If $p = 2$ and ϕ_1 is already determined, then there is only one direction orthogonal to ϕ_1, and we immediately obtain ϕ_2.
- If $p > 2$, there are infinitely many directions orthogonal to ϕ_1.
- PCA maximizes (4) with the additional constraint that ϕ_2 must be orthogonal to ϕ_1.

Third, fourth, and so on.

- Once ϕ_2 is computed, PCA finds ϕ_3 by maximizing (4) with the additional constraint that ϕ_3 has to be orthogonal to both ϕ_2 and ϕ_1.
- And so on to find ϕ_4, ϕ_5, etc.

Geometric interpretation: redux

Alternative interpretation: principal components provide low-dimensional linear surfaces that are closest to the observations.

- First PC loading vector is the line in p-dim space closest to the n obsns.
- First two PC loading vectors span the plane closest to the n obsns.
- First three PC loading vectors span the three-dim hyperplane closest to the n obsns. Etc.

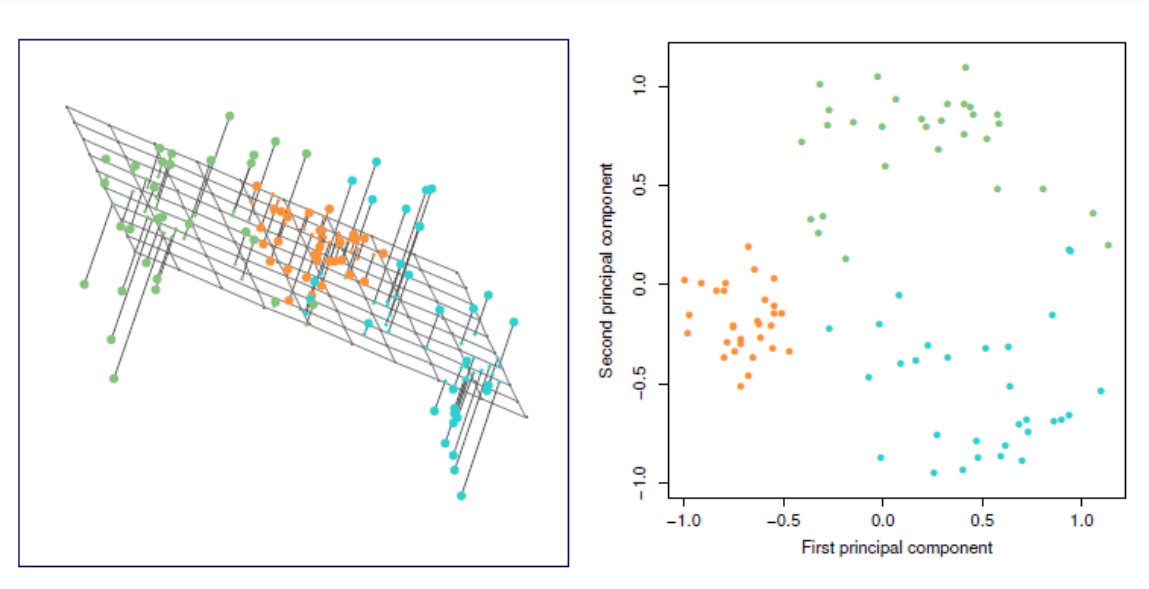

Figure 4: Figure by James et al. (2021). Observations simulated in three dimensions. The observations are displayed in color for ease of visualization. Left: The first two principal component directions span the plane that best fits the data (in sense of minimizing RSS). Right: The first two principal component score vectors give the coordinates of the projection of the observations onto the plane.

For any positive integer $M \leq \min\{n-1, p\}$, together the first M principal component score vectors and the first M principal component loading vectors provide an M-dimensional approximation to the ith observation:

$$x_{ij} \approx \sum_{m=1}^{M} z_{im}\phi_{jm}. \tag{5}$$

- How good is this approximation? It is the best, in the sense of the following optimization problem.
- Suppose we have a data matrix $\mathbf{X}$ that is column-centered, and have chosen some value of M.
- Of all possible approximations of the form $x_{ij} \approx \sum_{m=1}^{M} a_{im}b_{jm}$, consider which values $\{a_{im}\}$ and $\{b_{jm}\}$ minimize the sum of squares

$$\sum_{j=1}^{p}\sum_{i=1}^{n}\left(x_{ij} - \sum_{m=1}^{M} a_{im}b_{jm}\right)^2. \tag{6}$$

- It can be shown that for any value of M, the minimizers of (6) are exactly $a_{im} = z_{im}$ and $b_{jm} = \phi_{jm}$.
- When $M = \min\{n-1, p\}$, the representation (5) becomes exact.

The proportion of variance explained (PVE)

How much information is lost by projecting onto a few principal components, i.e., how much variance in the data is not contained in principal components?

- Assuming centered data, the *total variance* in the data is $\sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$, and the variance explained by the mth principal component is $\frac{1}{n} \sum_{i=1}^{n} z_{im}^2$.

$$\underbrace{\sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2}_{\text{Var. of data}} = \underbrace{\sum_{m=1}^{M} \frac{1}{n} \sum_{i=1}^{n} z_{im}^2}_{\text{Var. of first } M \text{ PCs}} + \underbrace{\frac{1}{n} \sum_{j=1}^{p} \sum_{i=1}^{n} \left(x_{ij} - \sum_{m=1}^{M} z_{im} \phi_{jm} \right)^2}_{\text{MSE of } M\text{-dimensional approximation}}$$

- *Proportion of the variance explained* (PVE) of the mth principal component is the quotient of these values, so

$$PVE_m = \frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2} . \tag{7}$$

- Proportion of variance explained by the first M principal components is

$$\sum_{m=1}^{M} PVE_m = 1 - \frac{RSS}{TSS} , \tag{8}$$

where TSS is total sum of squared elements of $\mathbf{X}$, and RSS is residual sum of squares of the M-dimensional PC approximation.

Screeplot

PVE is usually visualized with the *screeplot*.

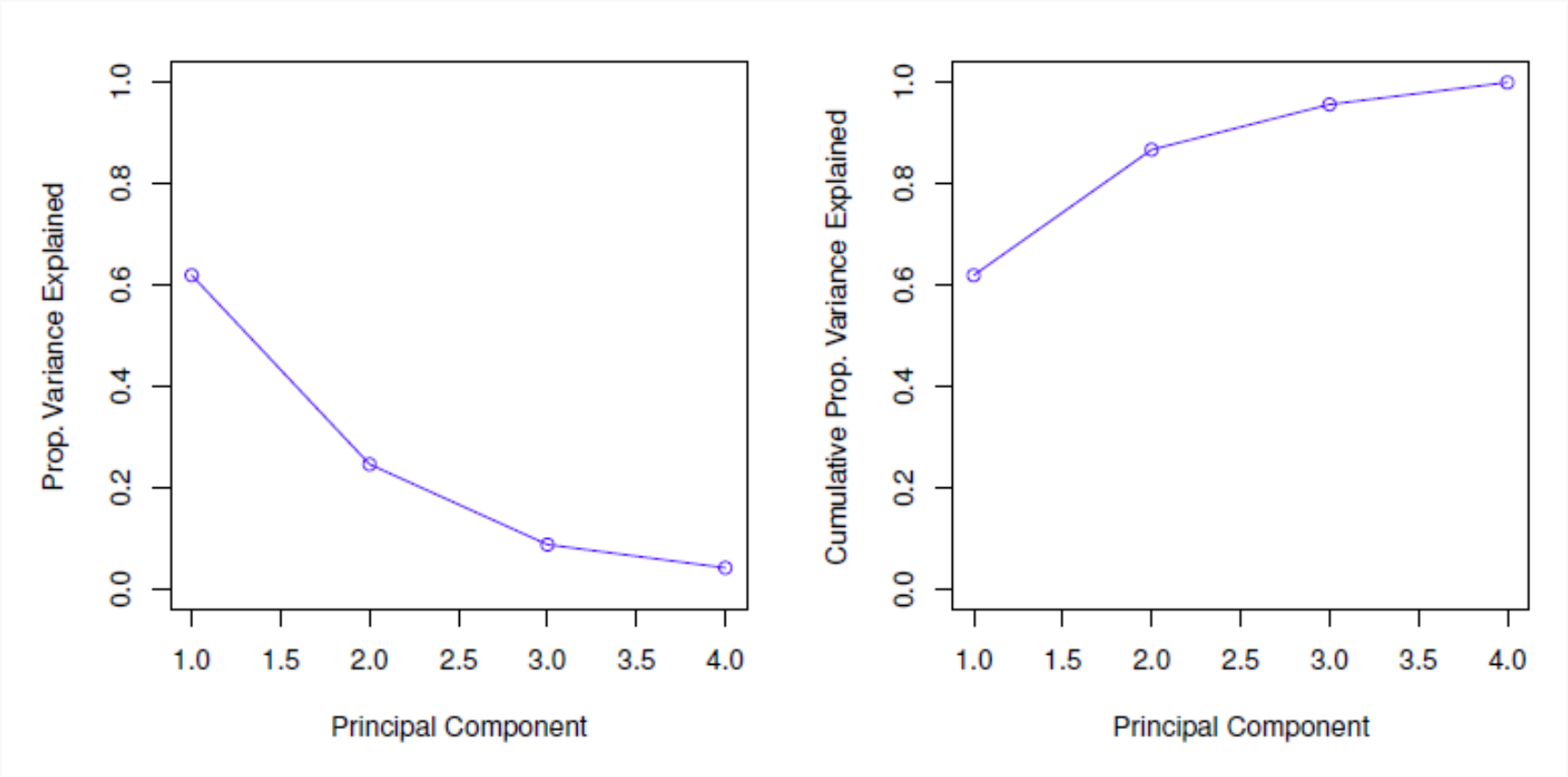

Figure 5: Figure by James et al. (2021). Left: A scree plot depicting the proportion of variance explained (PVE) by each of four principal components in USA`rrests` data. Right: The cumulative proportion of variance explained by four principal components.

Number of predictors

- Usually, we only want to have a small dimension in order to better visualize or understand the data.
- How many principal components do we need? How can we justify to use only three instead of four or more principal components (e.g.)?
  - There is no simple answer to this! There is no formula that can be applied universally that gives us the optimum value.
  - We can intuitively decide to choose the number of predictors by eyeballing the *screeplot* which depicts the proportion of variance explained (PVE).
  - In the screeplot, we look for a point at which the PVE by each subsequent principal component drops significantly off (this is subjective). Such a drop is often referred to as an *elbow* in the screeplot, and the rule by choosing the point is thus called *elbow rule*.

Scaling the variables

We are usually interested to analyze the impact of certain features in relation to their variation.

- The variance of features can be large solely because their values are.
- Not an issue if features are measured in the same units.
- So, by scaling (by their standard deviation), the variation among all predictors is comparable, independently of their magnitude.
- In general, scaling the variables to have standard deviation one is recommended.

Other uses for PCA

Many statistical techniques, such as regression, classification, and clustering, use the full $n \times p$ data matrix.

- Can instead use the $n \times M$ matrix whose columns are the first $M \ll p$ principal component score vectors.
- This can lead to less noisy results, since often the signal in a data set is concentrated in its first few principal components.