Principal Component Analysis II

PCA: An alternate technique for EDA and feature generation.

Today’s Roadmap

PCA Review

Interpreting PCA

Applications of PCA

PCA Review

[Recap] How to obtain Principal Components

a. Compute SVD on **centered** $\boldsymbol{X}$:

$$X = U\Sigma V^T$$

b. Take the first k columns of XV or $U\Sigma$

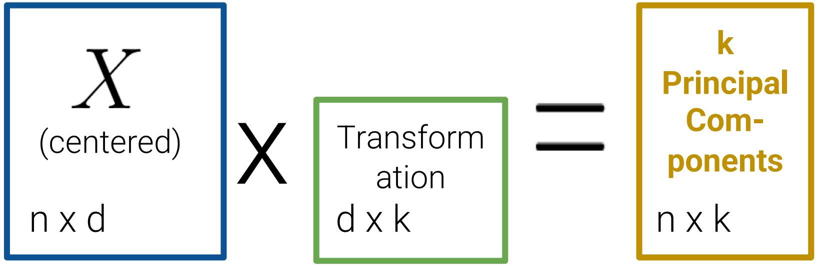
.

These are the first k **principal components**.

c. If we wanted to get a **rank-k approximation** of $\boldsymbol{X}$:
(σ_j: j-th singular value, $\mathbf{u}_j$: j-th col of U, v_j: j-th col of V)

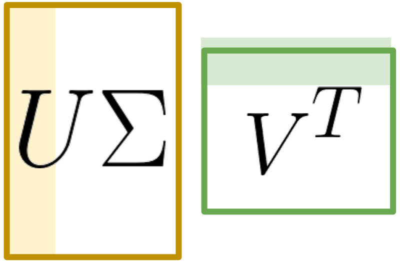
This effectively gives us k “features” from our d original features.

For most applications, we directly analyze the principal components themselves.




```
U[:, 0:k] @ np.diag(S[0:k]) @ Vt[0:k, :]
```

$$X_p = \sum_{j=1}^{p} \sigma_j u_j v_j^T$$



[Recap] How much variance is captured

- The principal components are a low-dimension representation that capture as much of the original data's **total variance** as possible.
- **Component scores** tell us how much variance each principal component captures:

$$i\text{-th component score} = \frac{(i\text{-th singular value})^2}{n}$$

- The component scores sum to total variance if we **center our data**.

PCA with SVD

To get the first **two principal components**, first find the SVD $X = U\Sigma V^T$.

100 x 4		4x2		100 x 2
X (centered)	×	Transformation	=	Principal Components 1 and 2

$$X \times V = U\Sigma$$

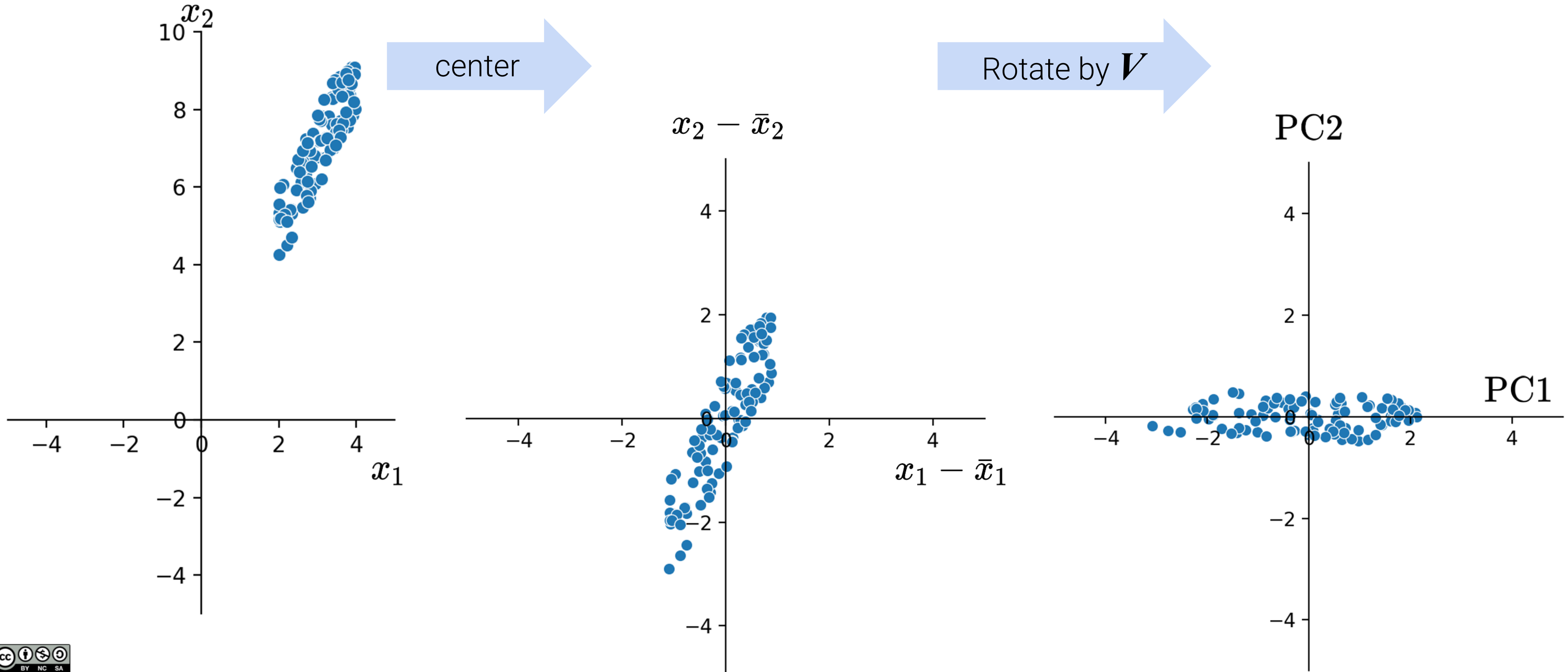
Take the first 2 **columns of $U\Sigma$** (or XV). These are PC1 and PC2.

why?

The first n **rows of V^T** are **directions** for the n principal components.

How PCA transforms data, visually

PCA first centers the data matrix, then rotates it such that the direction with the most variation (i.e. the direction that's the most spread-out) is aligned with the x-axis.



Columns of V are the directions

$$XV = \begin{bmatrix} | & & | & & | \\ \vdots & & \vdots & & \vdots \\ x_1 & \dots & x_j & \dots & x_d \\ \vdots & & \vdots & & \vdots \\ | & & | & & | \end{bmatrix} \begin{bmatrix} v_{11} & | & & | \\ \vdots & & & \\ v_{j1} & \dots & v_k & \dots & v_d \\ \vdots & & & \\ v_{d1} & | & & | \end{bmatrix}$$

$X \in \mathbb{R}^{n \times d}$

$V \in \mathbb{R}^{d \times d}$

transformation

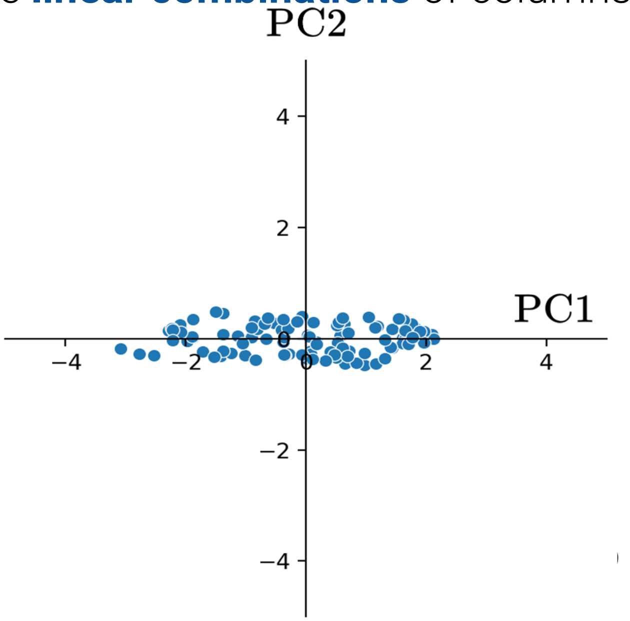
$$\underbrace{v_{11}}_{\text{scalar 1}} \cdot \underbrace{\begin{bmatrix} | \\ \vdots \\ x_1 \\ \vdots \\ | \end{bmatrix}}_{\text{feature 1}} + \cdots + v_{j1} \cdot \begin{bmatrix} | \\ \vdots \\ x_j \\ \vdots \\ | \end{bmatrix} + \cdots + v_{d1} \cdot \begin{bmatrix} | \\ \vdots \\ x_d \\ \vdots \\ | \end{bmatrix} = \text{PC1}$$

The elements of each **column of V (row of V^T)** **rotate** the original feature vectors into a principal component.

The first column of V indicates **how each feature contributes** (e.g., positively, negatively, etc.) to PC1.

Principal Components

- Principal components are all **orthogonal** to each other
  - Why? Recall that columns of U are orthonormal!
- Principal Components are **axis-aligned**
  - If we plot two PCs on a 2D plane, one will lie on the x-axis, the other on the y-axis
- Principal Components are **linear combinations** of columns in our data $\boldsymbol{X}$



Interpreting PCA

[From last time] Why perform PCA?

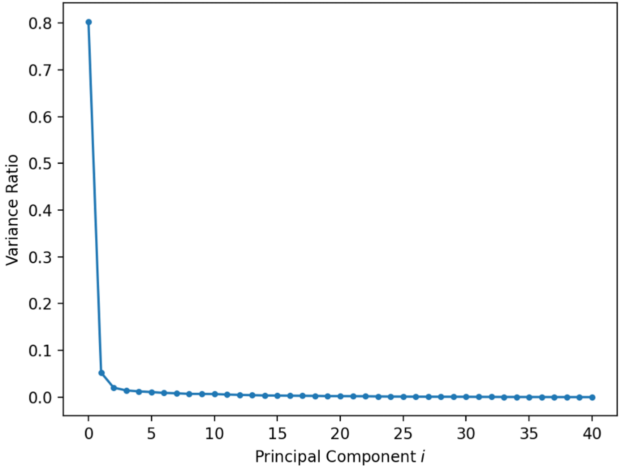
We often perform PCA during the Exploratory Data Analysis (EDA) stage of our data science lifecycle (if we already know what to model, we probably don't need PCA); it helps us with

- Visually identifying **clusters** of similar observations in high dimensions.
- Removing irrelevant dimensions if we suspect that the dataset is inherently low rank. For example, if the columns are **collinear**: there are many attributes but only a few mostly determine the rest through linear associations.
- Finding a small basis for representing variations in complex things, e.g., images, genes.
- Reducing the number of dimensions to make some computation cheaper.

Scree Plot

If the first two singular values are large and all others are small, then **two dimensions are enough** to describe most of what distinguishes one observation from another. If not, then a PCA scatter plot is omitting lots of information.

A **scree plot** shows the variance ratio captured by each principal component, largest first.



Scree [wikipedia]

Biplot

$$\underset{\text{scalar 1}}{v_{11}} \cdot \begin{bmatrix} | \\ \vdots \\ x_1 \\ \vdots \\ | \end{bmatrix} + \cdots + v_{j1} \cdot \begin{bmatrix} | \\ \vdots \\ x_j \\ \vdots \\ | \end{bmatrix} + \cdots + v_{d1} \cdot \begin{bmatrix} | \\ \vdots \\ x_d \\ \vdots \\ | \end{bmatrix} = \text{PC1}$$

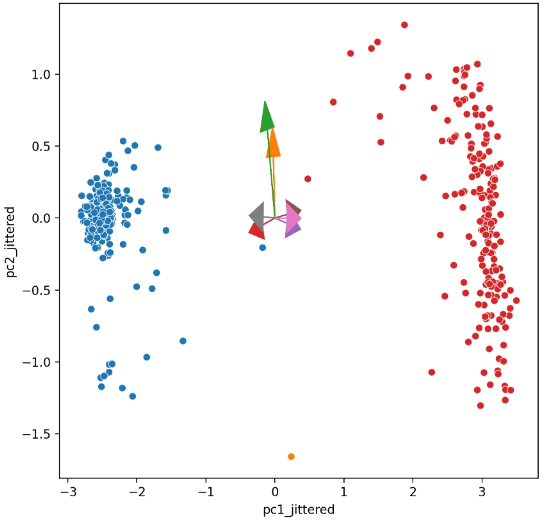
The $\boldsymbol{i}$-th column of $\boldsymbol{V}$ indicates **how each feature contributes** to PC $\boldsymbol{i}$.

Biplots superimpose the **directions** onto the plot of PC1 vs. PC2.

Vector j corresponds to the direction for feature j, e.g., (v_{1j}, v_{2j}).

- There are several ways to scale biplots vectors; in this course we plot the direction itself.
- For other scalings, which can lead to more interpretable directions/loadings, see [SAS biplots].

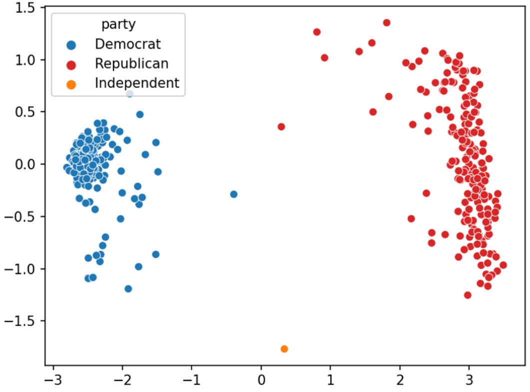
Through biplots, we can interpret how features correlate with the principal components shown: positively, negatively, or not much at all.



Summary: Plots based on PCA

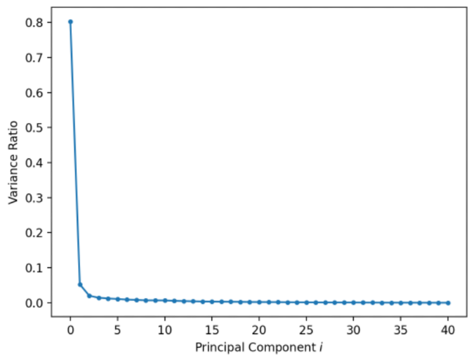
PCA Plot

Scatter plot of PC1 against PC2
- Helps us assess similarities between our data points and if there are any clusters in our dataset.



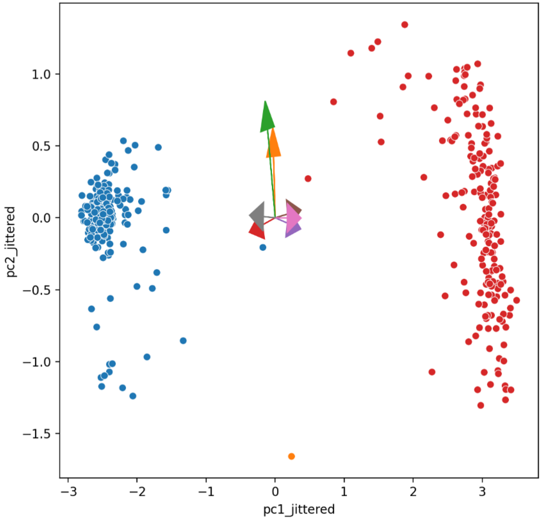
Scree Plot

Line plot showing the **variance ratio** captured by each principal component, largest first.
- If first two is large enough, we know PCA plot is good representation of data
- “Elbow” method to assess how many PCs to use



Biplot

PCA plot + **directions** of feature importance for PC1 and PC2
- All benefits from PCA plot, and
- Shows how much some features contribute to PC1/2

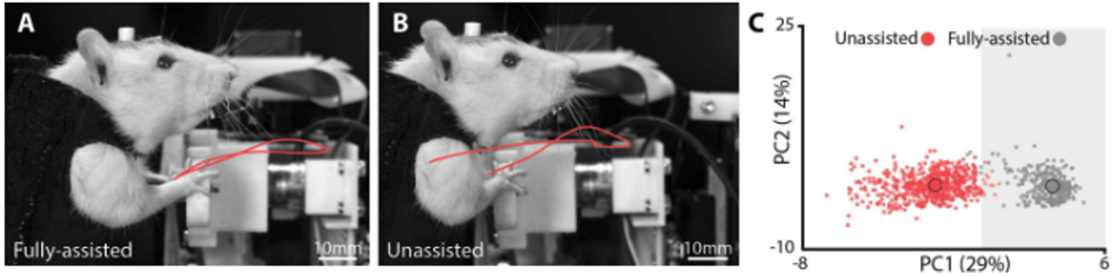


Applications of PCA

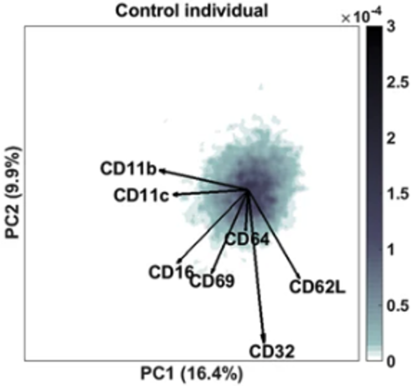
PCA in Biology

PCA is commonly used in biomedical contexts, which have many named variables!

1. To cluster data (Paper 1, Paper 2)



2. To identify correlated variables (interpret rows of V^T as linear coefficients) (Paper 3). Uses biplots.



Why perform PCA?

We often perform PCA during the **Exploratory Data Analysis** (EDA) stage of our data science lifecycle (if we already know what to model, we probably don't need PCA); it helps us with

- Visually identifying clusters of similar observations in high dimensions.
- Removing irrelevant dimensions if we suspect that the dataset is inherently low rank. For example, if the columns are collinear: there are many attributes but only a few mostly determine the rest through linear associations.
- Finding a small basis for representing variations in complex things, e.g., images, genes.
- Reducing the number of dimensions to make some computation cheaper.

Demo

Image Classification

Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Han Xiao, Kashif Rasul, Roland Vollgraf. arXiv:1708.07747
https://github.com/zalandoresearch/fashion-mnist

In **machine learning**, **PCA** is often used as a **preprocessing step** prior to training a supervised model..

Lecture 23

Principal Component Analysis II

Content credit: Acknowledgments