Dr. Denton Bobeldyk

CIS 365 Artificial Intelligence

Dimensionality Reduction

Delivery Methods

Lecture

Videos

Lab Time

Small Groups

Dimensionality Reduction

- Defined as reducing the number of features in a dataset while retaining its essential information
- Curse of Dimensionality: challenges associated with high dimensional data

Dimensionality Reduction Uses

- Data visualization
- Noise reduction
- Computational efficiency

Curse of Dimensionality Demonstration
Exercise (10-15 minutes)

Using Python:

1. Randomly generate 10 numbers using ‘np.random.rand’

2. Measure the average distance between all 10 of the data points using the euclidean distance (pdist)

3. Increase the number of features for the randomly generated data to 2 features. Repeat for 5 and then 10.

Plot out the resulting distance, if time permits, continue to increase the dimensions.

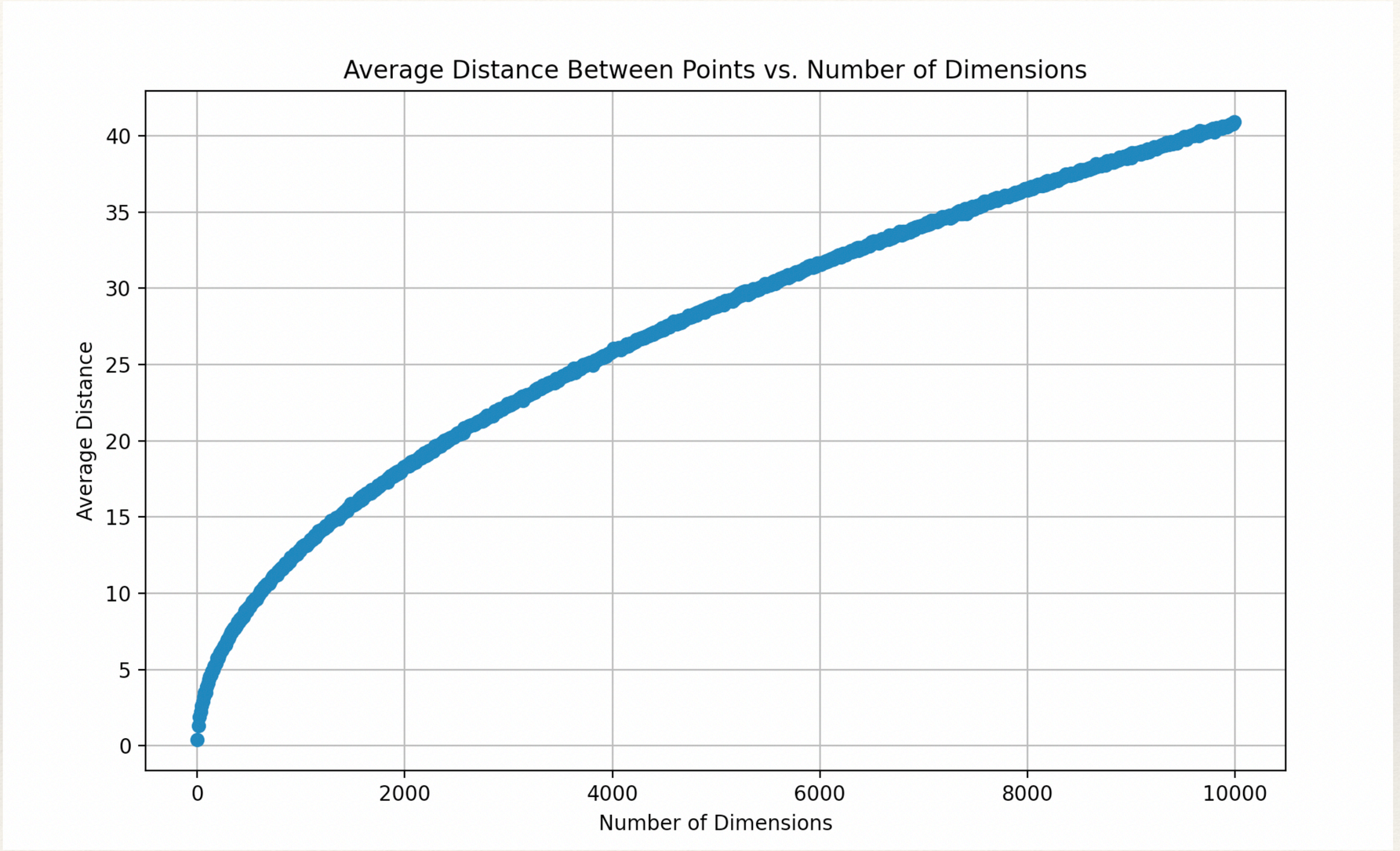
The following python fragments may be useful:

```
import numpy as np
from scipy.spatial.distance import pdist

np.random.rand(numberOfPoints, numberOfFeatures)
distances = pdist(data)

np.mean(distances)
```

Graph next slide


Average Distance Between Points vs. Number of Dimensions
Average Distance
Number of Dimensions
0
5
10
15
20
25
30
35
40
0
2000
4000
6000
8000
10000

Impact on Machine Learning

- Distance Metrics: As dimensions increase, the distance between the points grow, making it harder for algorithms like kNN to find meaningful neighbors
- Sparsity: In high dimensional space, the data becomes sparse, which makes it harder for models to detect patterns, detect correlations, increases the risk of overfitting. It also causes problems for clustering, classification and regression.
- Visualization: In low dimensions (2D or 3D), you can plot data and visually understand clusters or patterns, but in high dimensions this becomes nearly impossible

Methods to reduce dimensions

- Principal Component Analysis
- Autoencoders
- t-SNE

Feature Extraction vs Feature Selection

- Extraction: Creating new features (e.g., PCA)
- Selection: Picking a subset of existing features (e.g., removing irrelevant features)

Feature Extraction vs Feature Selection

- Feature Selection Examples:
  - Choosing furniture for your apartment

Methods to Reduce Dimensions (Feature Extraction)

- Principal Component Analysis
- Autoencoders
- t-SNE

Methods to Reduce Dimensions (Feature Extraction)

- Principal Component Analysis
- Autoencoders
- t-SNE

Principal Components Analysis

- PCA Overview
- PCA Applications
- PCA - What is it? How can we calculate?

PCA Overview

- Curse of Dimensionality
  - Image resolutions continue to increase
  - Feature vectors representing a social media identity

PCA Overview

- Curse of Dimensionality
  - Image resolutions continue to increase
  - Feature vectors representing a social media identity

Is it possible to represent each of these in a more compact way?

PCA Overview

- Curse of Dimensionality
  - Image resolutions continue to increase
  - Feature vectors representing a social media identity
  - Spatial Example
  - Spatial Example - Classification

PCA Overview

- PCA can be used to compactly represent features

PCA Overview

- White Board work: Athlete Classes

PCA Overview

- Goal is to maximize variance along the new axes (principal components)
- Components with higher variance capture more information

PCA Math Overview

- Vector Representation:
  - $\overrightarrow{x} = [x_1, x_2, \ldots, x_n]$
  - Each ‘x’ in the vector represents a feature

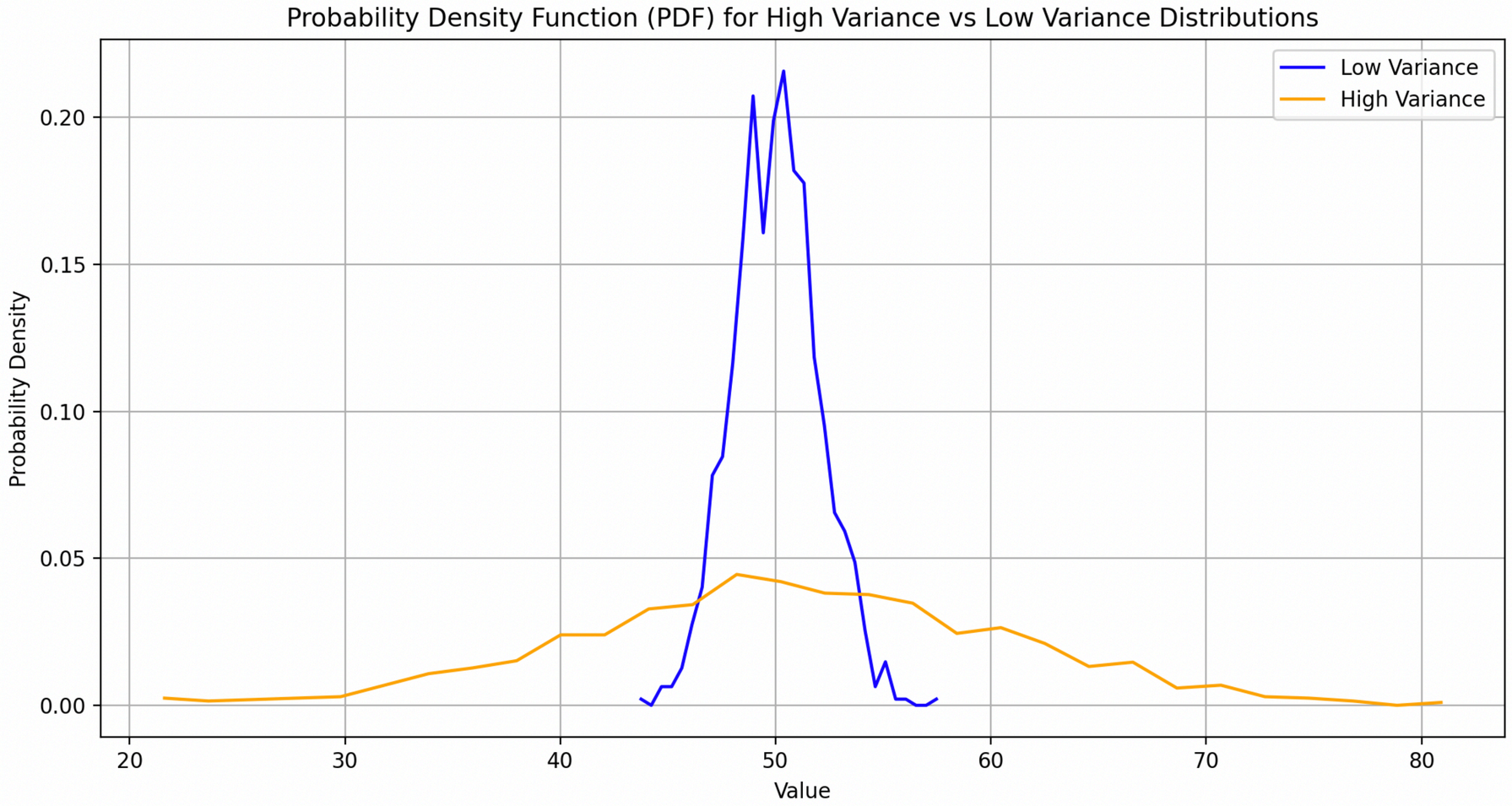
Variance

Variance: measure of how much the values in the dataset differ from the mean of that dataset.

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Variance is the square of the standard deviation

Variance


Probability Density Function (PDF) for High Variance vs Low Variance Distributions
Low Variance
High Variance
Probability Density
0.00
0.05
0.10
0.15
0.20
20
30
40
50
60
70
80
Value

Covariance

Covariance measures how two features change together

* If two features increase together, there covariance is positive

* If one increases while the other decreases the covariance is negative

Covariance Matrix - Review

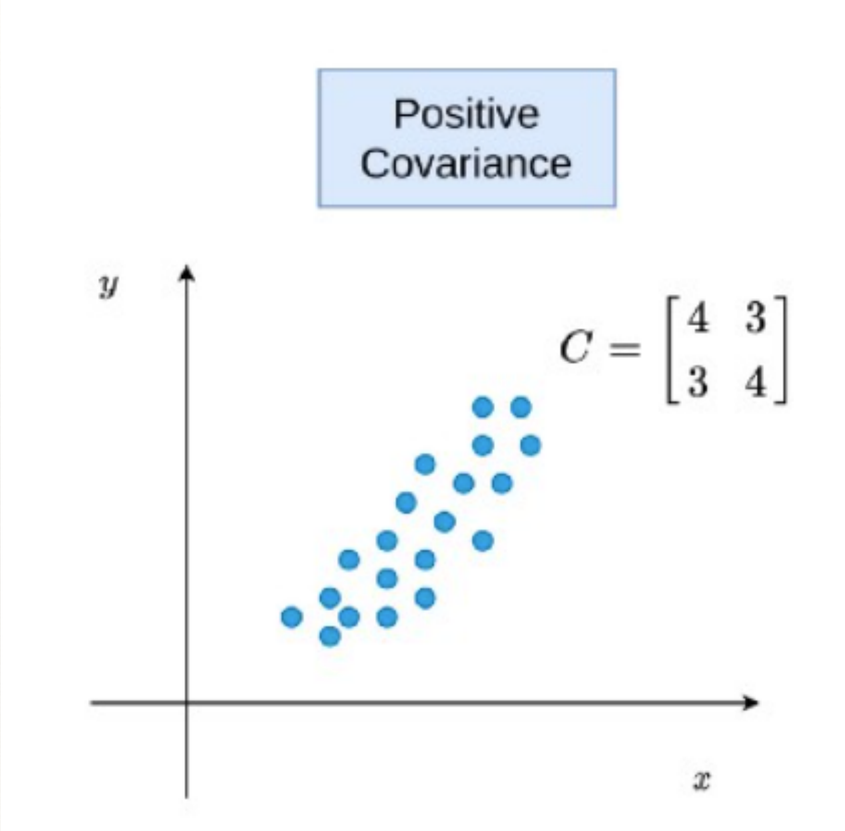


Image from: https://towardsdatascience.com/5-things-you-should-know-about-covariance-26b12a0516f1

Covariance Matrix - Review

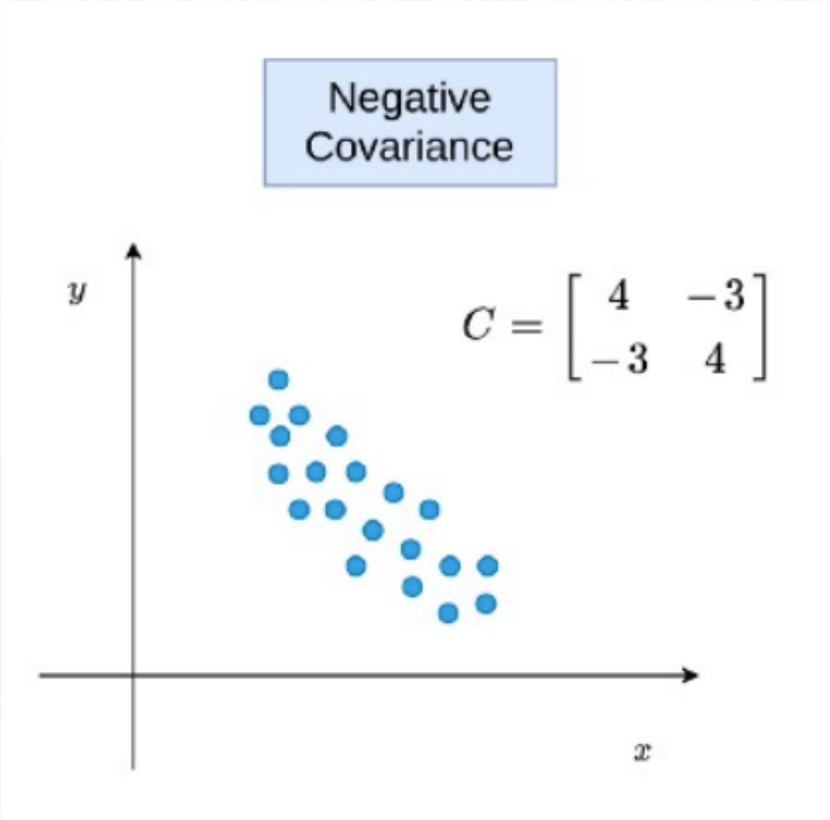


Image from: https://towardsdatascience.com/5-things-you-should-know-about-covariance-26b12a0516f1

Covariance Matrix - Review

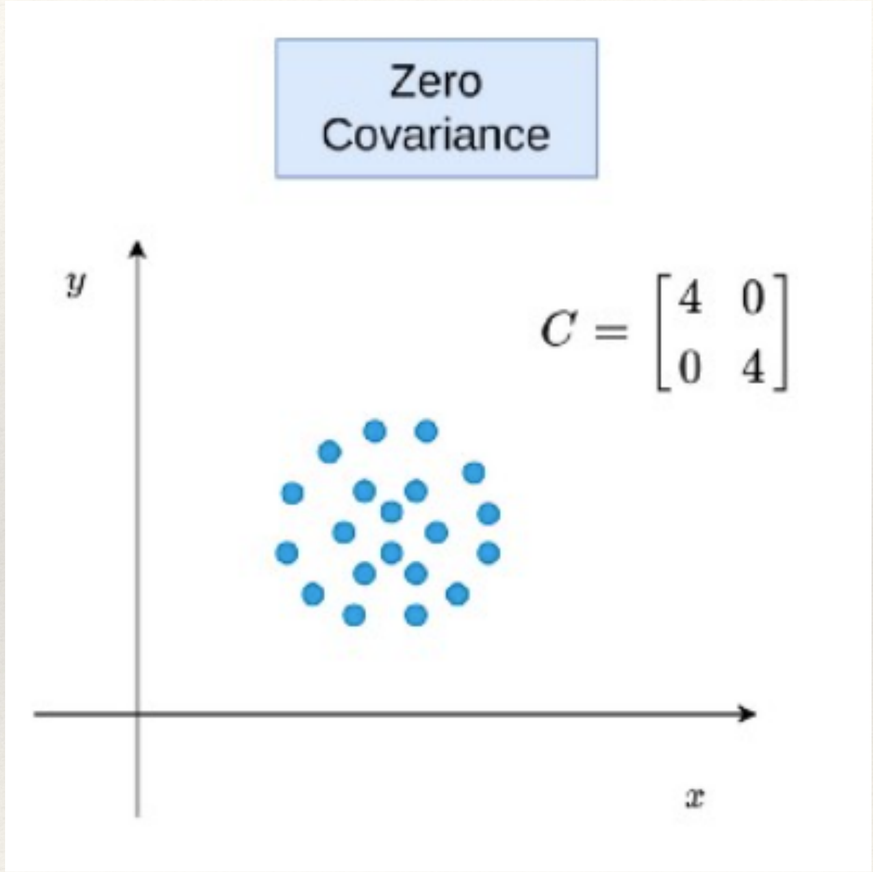
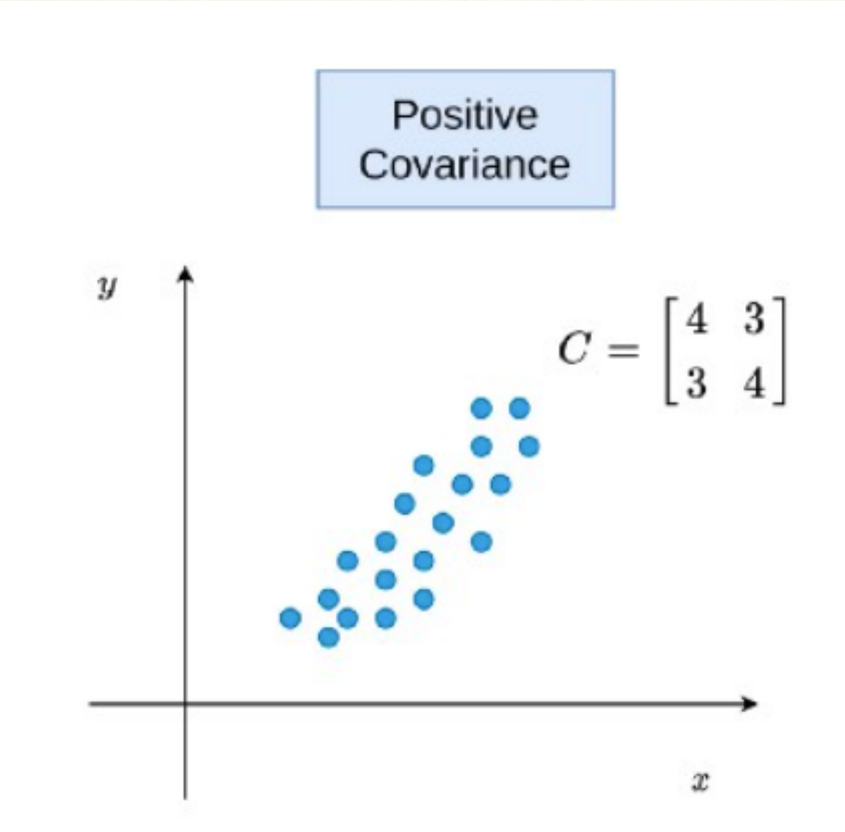


Image from: https://towardsdatascience.com/5-things-you-should-know-about-covariance-26b12a0516f1

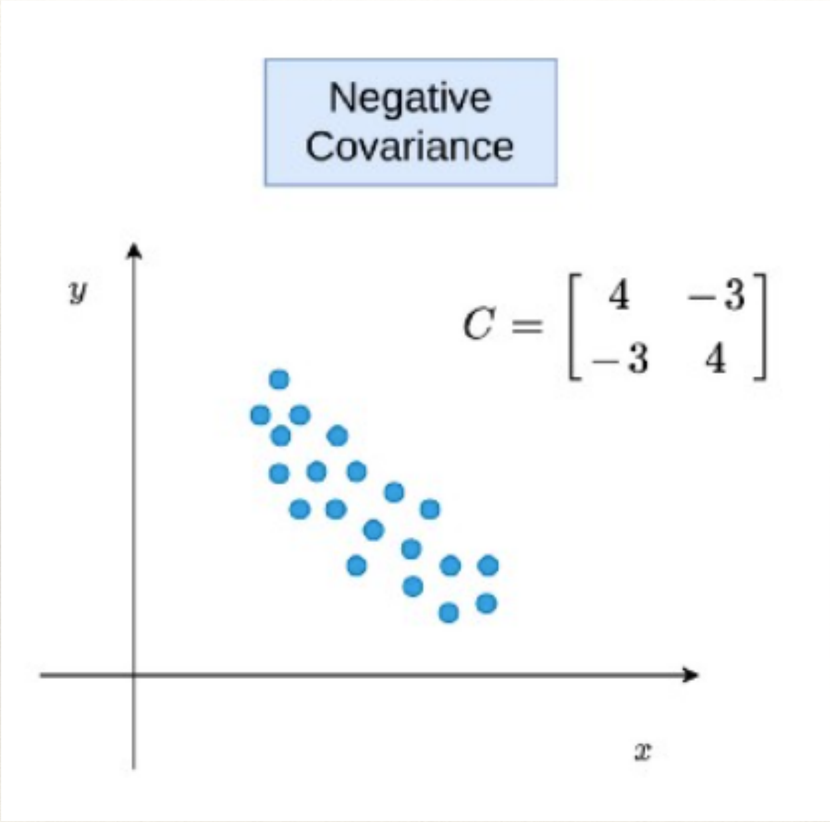
Covariance Matrix - Review



Give an example of two features that might have a positive covariance

Image from: https://towardsdatascience.com/5-things-you-should-know-about-covariance-26b12a0516f1

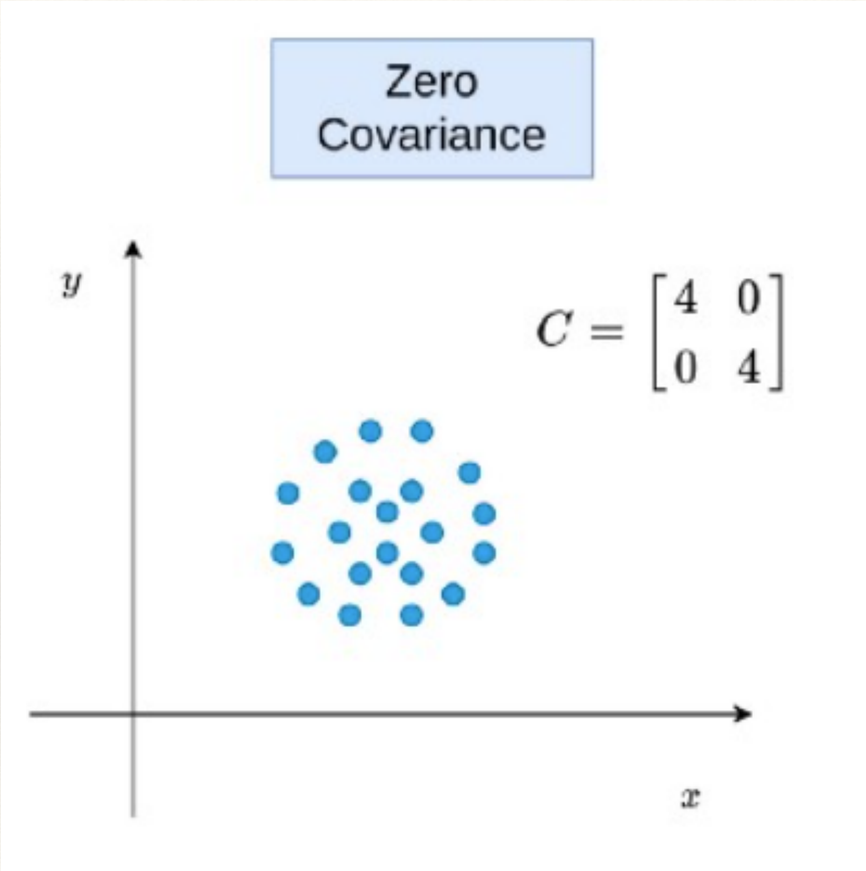
Covariance Matrix - Review



Give an example of two features that might have a negative covariance

Image from: https://towardsdatascience.com/5-things-you-should-know-about-covariance-26b12a0516f1

Covariance Matrix - Review



Give an example of two features that might have a zero covariance

Image from: https://towardsdatascience.com/5-things-you-should-know-about-covariance-26b12a0516f1

Covariance Matrix - Review

$$\mathrm{Cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})$$

Cov(X,Y) represents the covariance between variable X and Y

n is the number of data points

$\bar{X}$, $\bar{Y}$ are the means of X and Y respectively

Calculates how much two variables vary together compared to their individual means

Covariance Matrix - Review

$$\text{Cov}(X) = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \text{Cov}(X_1, X_3) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \text{Cov}(X_2, X_3) \\ \text{Cov}(X_3, X_1) & \text{Cov}(X_3, X_2) & \text{Var}(X_3) \end{bmatrix}$$

Covariance Matrix - Review

$$\text{Cov}(X) = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \text{Cov}(X_1, X_3) & \text{Cov}(X_1, X_4) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \text{Cov}(X_2, X_3) & \text{Cov}(X_2, X_4) \\ \text{Cov}(X_3, X_1) & \text{Cov}(X_3, X_2) & \text{Var}(X_3) & \text{Cov}(X_3, X_4) \\ \text{Cov}(X_4, X_1) & \text{Cov}(X_4, X_2) & \text{Cov}(X_4, X_3) & \text{Var}(X_4) \end{bmatrix}$$

Covariance Matrix - Python Exercise

What sort of options does Python have to find the COV?

Some solutions next slide

Covariance Matrix - Numpy

```python
import numpy as np

# Sample data with 4 features (rows are samples, columns are features)
data = np.array([
    [4.0, 2.1, 3.5, 7.1],
    [4.2, 2.3, 3.8, 7.4],
    [3.9, 2.0, 3.7, 7.0],
    [4.1, 2.2, 3.6, 7.2],
    [4.3, 2.4, 3.9, 7.5]
])

# Calculate the covariance matrix using numpy
cov_matrix = np.cov(data, rowvar=False)
print("Covariance Matrix using SciPy:")
print(cov_matrix)
```

Covariance Matrix - Pandas

```python
import pandas as pd

# Sample data in a DataFrame
df = pd.DataFrame({
    'Feature1': [4.0, 4.2, 3.9, 4.1, 4.3],
    'Feature2': [2.1, 2.3, 2.0, 2.2, 2.4],
    'Feature3': [3.5, 3.8, 3.7, 3.6, 3.9],
    'Feature4': [7.1, 7.4, 7.0, 7.2, 7.5]
})

# Calculate the covariance matrix
cov_matrix = df.cov()
print("Covariance Matrix:")
print(cov_matrix)
```

Eigenvalues and Eigenvectors

Eigenvectors define the directions of the new axes (principal components)
Eigenvalues indicate the magnitude (amount of variance) along those directions

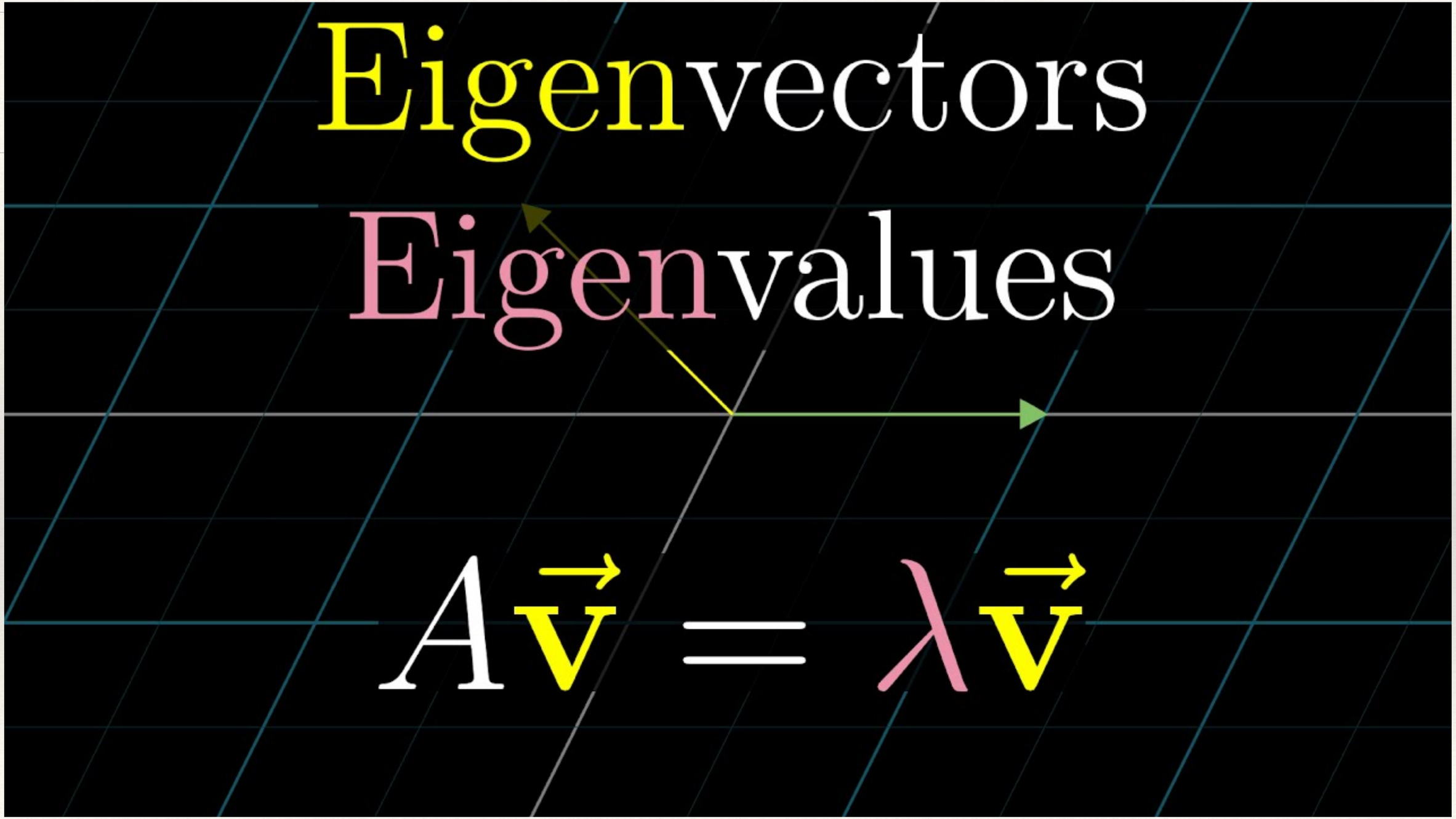
Eigenvalues and Eigenvectors

- Eigenvalues are sorted in descending order and the corresponding eigenvectors are used to form the principal components
- Select the top k components that capture the most variance

PCA Steps

- Standardize the data
  - Each feature has a mean of 0 and a standard deviation of 1
- Calculate the covariance matrix
- Compute the Eigenvalues and Eigenvectors of the covariance matrix
- Sort the Eigenvalues and Eigenvectors
- Select the top k principal components

Eigenvectors
Eigenvalues



$$A\vec{\mathbf{v}} = \lambda\vec{\mathbf{v}}$$

END