Recitation 7: Unsupervised Machine Learning

CIS 5450: Big Data Analytics
Fall 2023
October 27, 2023

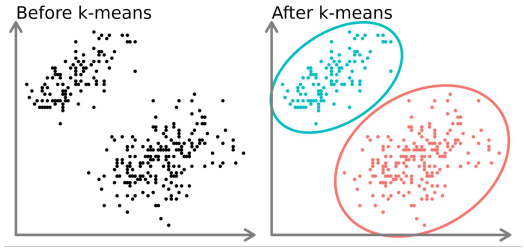
Penn Engineering
University of Pennsylvania

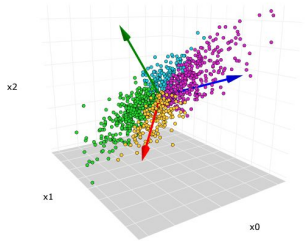
Agenda:

1. What is unsupervised learning?
2. Principal Component Analysis (PCA)
3. K-Means Clustering

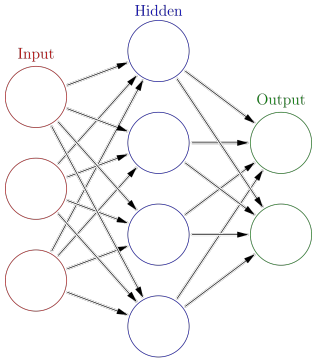
The Types of Machine Learning

Unsupervised: Find **structure** in data





Supervised: Function mapping **inputs to outputs**



Penn Engineering
University of Pennsylvania

PCA Motivation:

Imagine you are trying to run a machine learning algorithm on…

1. Very noisy data
2. High-dimensional image data (for image classification)
3. Data with a large number of features (10,000+)

The Problem: How do we determine which features matter, and which are irrelevant or noisy?

Principal Component Analysis:

Overview of Principal Component Analysis (PCA)

1. *Unsupervised* algorithm
2. Takes in: a matrix of data points and their respective features
3. Outputs: a new set of orthogonal (and independent) feature vectors, *ordered* by their relative importance

How does this solve the problems previously?

1. Denoising: unimportant “noisy” features are discarded
2. Dimensionality Reduction: we can keep only the most important features - this reduces the dimensionality and size of our data
3. Scalable to massive quantities of data

A Conceptual Overview

Idea: Express our data in terms of a new set of vectors.

- From the x- and y- axes
- To the vectors in the ellipse

Why? Because these vectors represent the direction of *maximum variance*.

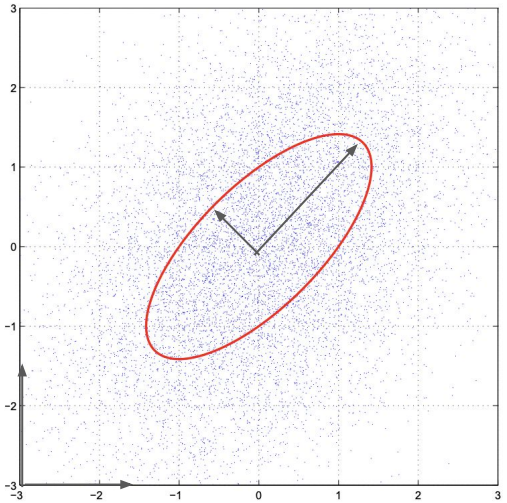


Image Source: ESE 2240

Penn Engineering
University of Pennsylvania

How to Perform PCA

Step One: Scale the data by subtracting the mean and dividing by the standard deviation of each feature.

$$x_{new} = \frac{x - \mu}{\sigma}$$

Step Two: Compute the Covariance Matrix.

$$\begin{bmatrix} E[(X_1 - E[X_1])(X_1 - E[X_1])] & \cdots & E[(X_1 - E[X_1])(X_p - E[p])] \\ \vdots & \ddots & \vdots \\ E[(X_p - E[X_p])(X_1 - E[X_1])] & \cdots & E[(X_p - E[X_p])(X_p - E[X_p])] \end{bmatrix}$$

Step Three: Take the eigenvectors and eigenvalues of the matrix.

Step Four: Project your data onto the eigenvectors/values.

Step Five: Select the number of principal components you want to keep, and discard the rest of those in your data.

Image Sources: Medium, CIS 5450 Lecture Slides

Penn Engineering
University of Pennsylvania

An Important Note:

PCA is NOT Scale Invariant!

What does this mean?

If we rescale different features, we change the **variance** of those features.

This gives us different results.

Therefore, PCA always requires normalization before you perform it!

*For a more in-depth mathematical explanation of this, we'd be happy to chat after class.

Penn Engineering
University of Pennsylvania

Pros and Cons of PCA

Pros:

- We need less memory once we reduce dimensions
- Reduces noise

Cons:

- We may sometimes lose the explainability of features

Clustering:

- Data generally has hidden patterns
- Data may not always come with labels, but it may still contain interesting information
- **Applications:**
  - Customer segmentation (for marketing and recommendation)
  - Search result clustering
- **Different types of clustering:**
  - Hierarchical: Iteratively identify closest clusters and keep merging. Start with each point being a cluster in itself. (https://www.displayr.com/what-is-hierarchical-clustering)
  - K-means: Iteratively find points closest to centroids and calculate new centroids

K-Means Clustering:

Overview of K-Means Clustering

1. *Unsupervised* algorithm
2. Takes in: a matrix of data points and their respective features, as well as the desired number of clusters
3. Outputs: a specified number of cluster centers and assigns each data point to the nearest cluster center.

How to Perform K-Means

Step One: Fix k, the number of centroids.

Step Two: Assign each point to the centroid to which it is closest in distance.

Step Three: Re-calculate the cluster centroids from the points assigned to each respective cluster from Step Two.

Step Four: Repeat Steps Two & Three until convergence (when the centroids no longer change).

Step Five (optional): Repeat all of the above steps for different k's to find the optimal k.

Pros and Cons of K-Means Clustering

Pros:

- Relatively simple and fast to implement
- Does not require labeled data (unsupervised learning)

Cons:

- Have to choose ‘k’ manually
- Final clusters dependent on initialization
- Sensitive to outliers

Demo

https://colab.research.google.com/drive/1mhemCPlTF2WbJsUFlSHgheReDs5sdU0K?usp=sharing

Performing this in Scikit-Learn:

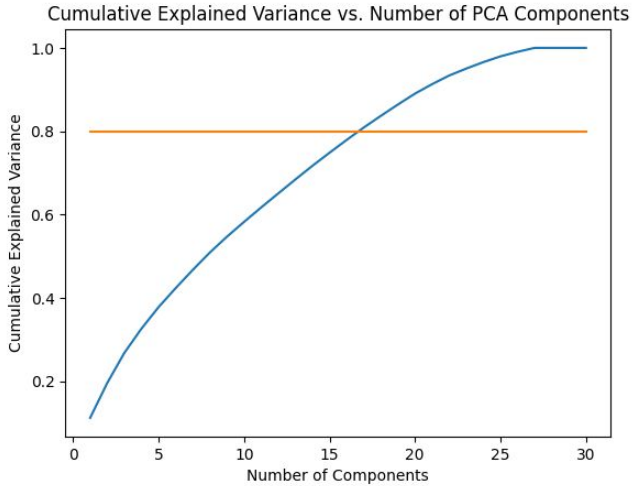
```python
# PCA Imports
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Normalization to address the fact that PCA is not scale-invariant
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Instantiate and Fit PCA
pca = PCA()
X2 = pca.fit(X_train_scaled)

# Intermediate Step: Identify optimal number of principal components using cumulative variance
# (see the graph)
explained_variance_ratios = pca.explained_variance_ratio_
cum_evr = np.cumsum(explained_variance_ratios)

# Redefine and refit PCA
n = 17
pca = PCA(n_components = n)
X_train_pca = pca.fit_transform(X_train_scaled)
X_test_pca = pca.transform(X_test_scaled)
```



Penn Engineering
University of Pennsylvania