Big Data for Public Policy

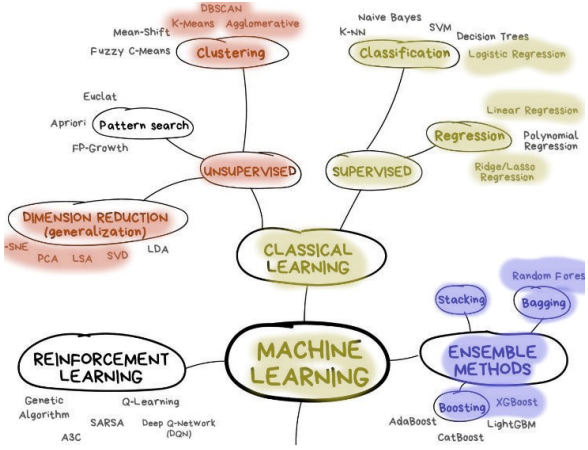
Unsupervised ML

Sergio Galletta

ETHZ Zurich

27/04/2023

What we will do today



Unsupervised Learning

- **Unsupervised learning** is a type of machine learning where the goal is to discover patterns in data **without any labeled** examples
- Unlike supervised learning, there are no target variables to predict, and the algorithm must find patterns and structure in the data on its own
- It can be used for tasks such as clustering, anomaly detection, and dimensionality reduction.

Dimensionality reduction

- Features $X_1, X_2, ..., X_p$ measured on observations, but no associated labeled variable
- **Why do we need dimensionality reduction?**
  - ML problems often involve thousands of features.
  - Especially in the case of text data.
  - Need for computational tractability and finding a good solution
- Can be used as a descriptive tool.
  - Extract information from the data and visualize it.
  - Discover subgroups among the variables or the observations
- Examples
  - Dimension reduction for pre-processing
  - Costumer segmentation in marketing

Principal Component Analysis (PCA)

- PCA is a technique for reducing the dimensionality of high-dimensional data
- Identifies the axis that accounts for the largest amount of variance in the data
- Finds a second axis, orthogonal to the first, that accounts for the largest amount of the remaining variance, and so on
- The unit vector defining the i^{th} axis is called the i^{th} principal component.

Principal Component Analysis (PCA) - Objectives

- Summarize a large set of feature variables with a smaller number of representative variables
  - collectively explains most of the variability in the original dataset
- Find a low-dimensional representation of the data that captures as much of the information possible
- If we can obtain a 2-dimensional representation, we can plot the observations in 2D.

Principal Component Analysis

What are we after?

- Each of the dimensions found by PCA is a linear combination of the p features
- The first principal component of a set of features $X_1, X_2, ..., X_p$ is the normalized linear combination of the features:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + ... + \phi_{p1}X_p$$

- ...that has the largest variance
- Normalized means that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$
- $\phi_1 = (\phi_{11}, \phi_{21}, ..., \phi_{p1})^T$ is the **loading vector** of the first principal component
- **The loading vector** represents the weights of the original variables that make up each principal component.

Principal Component Analysis - Computing the First PC

Maximizing the Sample Variance of Z_1:

- We want to find the values of $\phi_{11}, \phi_{21}, ..., \phi_{p1}$ that maximize the sample variance of Z_1, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$
- We can write the optimization problem as:

$$\max_{\phi_{11},\phi_{21},...,\phi_{p1}} \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2$$

$$\text{subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Principal Component Analysis - Computing the First PC

- We can rewrite the objective function as:

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

  where z_{i1} is the ith observation's value for the first principal component, and $z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + ... + \phi_{p1}x_{ip}$.
- Since the data has mean zero, we have $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$
- Using eigen decomposition (outside the scope of the class)
- $z_{11}, ..., z_{n1}$ are the **scores** of the first principal component
- The **score** represents the contribution of each observation to each principal component
- Solved using Singular Value Decomposition (SVD) [a standard linear algebra tool]

Principal Component Analysis - Computing the Second PC

Second Principal Component

- We can get the second principal component by finding the linear combination of the features that has the largest variance, subject to the constraint that it is orthogonal to the first principal component
- Z_2 is the linear combination of $X_1, X_2, ..., X_p$:

$$Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + ... + \phi_{p2}X_p$$

- Z_2 has maximal variance out of all linear combinations uncorrelated with Z_1
- To ensure that the second principal component is orthogonal to the first principal component, we need to add the constraint that:

$$\Phi_1^T \Phi_2 = 0$$

Principal Component Analysis - Projection on a 2D space

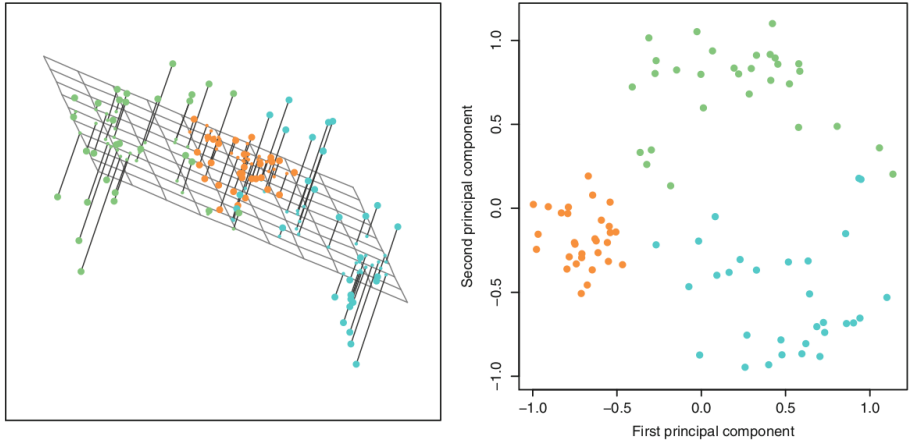


Figure 1: Illustration in 3D, projected on a 2D space.

- **Left:** Simulated data in 3 dimensions.
- **Right:** Projection on the first two principal components (plane represented on the left).

Principal Component Analysis - Pre-processing the variables

- Variables should:
  - be centered, to have mean zero
  - have the same variance 1
- the results obtained depend on whether the variables have been individually scaled
- Step done by default in python

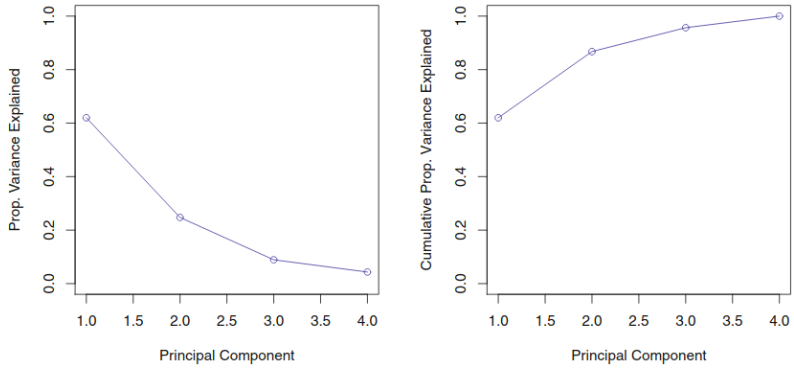
```
from sklearn.decomposition import PCA
pca = PCA(n_components=10)
X_train_pca = pca.fit_transform(X_train)
```

Principal Component Analysis - Proportion of the Variance Explained

- PVE (Proportion of the Variance Explained) measures how much of the information in a given data set is lost by projecting the observations onto the first few principal components
- To visualize PVE, we can plot the proportion of the variance explained by each principal component
- The PVE for the m^{th} principal component is defined as:

$$PVE_m = \frac{\text{Variance explained by the } m^{th} \text{ component}}{\text{Total variance}}$$

Principal Component Analysis - Proportion of the Variance Explained



- **Left:** proportion of variance explained by each of the four principal components
- **Right:** the cumulative proportion of variance explained by the four principal components

Principal Component Analysis -Choosing the Number of Dimensions

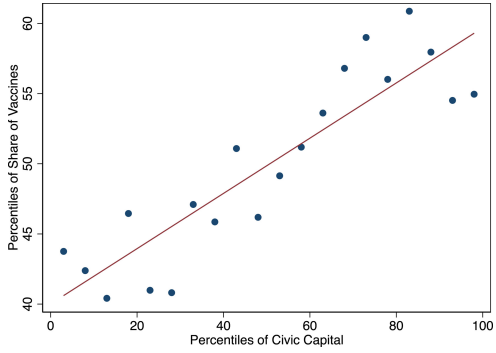
No criteria for deciding how many principal components (PC) are required, but some rules of thumb:

- Choose the smallest number of PC required to explain a sizable amount of the variation in the data
- For dimensionality reduction:
  - Explaining 95% of the variance is a good objective.
- For data visualization:
  - Focus on a small number of axes that you can interpret.
  - Do not interpret the components explaining less than 10%.

Example

The Role of Civic Capital on Vaccination - P Buonanno, S Galletta and M Puca

- Does civic capital affect vaccination rate?
- Focus on Lombardy municipality
- **Share of vaccinated individuals** for each municipality in Lombardy from June 1 to October 1, 2021
- **Civic capital** is proxy by the 1st PC of *organ donations*, the *share of urban solid waste recycling*, and *tax compliance*



Clustering Methods

- Clustering refers to a very broad set of techniques for finding subgroups, or clustering clusters, in a data set.
- When performing clustering, the aim is to group data into subsets so that
  - The objects grouped in each subset are similar, close to one another, homogeneous
  - And different from the objects in other groups

$\Rightarrow$ Find some structure in the data.

K-means Clustering

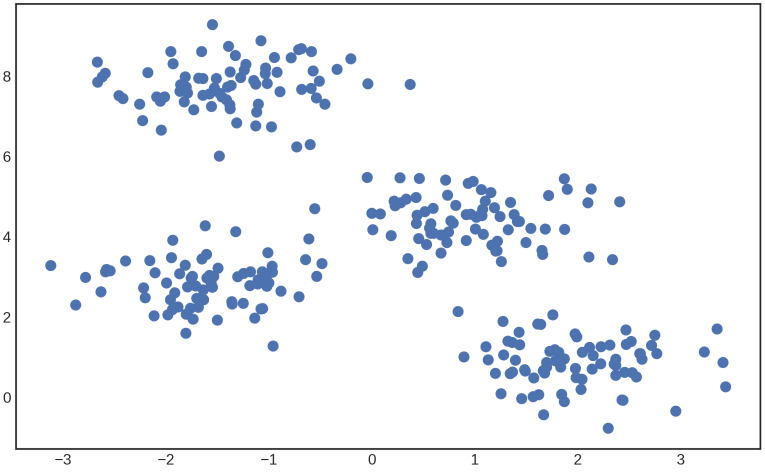
What is K-means Clustering?

- K-means clustering is a popular unsupervised machine learning algorithm used for partitioning data into a pre-specified number (k) of clusters
- The partitioning corresponds to an optimization problem that consists of:
  - Partitioning the data into k clusters of equal variance.
  - Minimizing the within-cluster sum-of-squares (**inertia**):

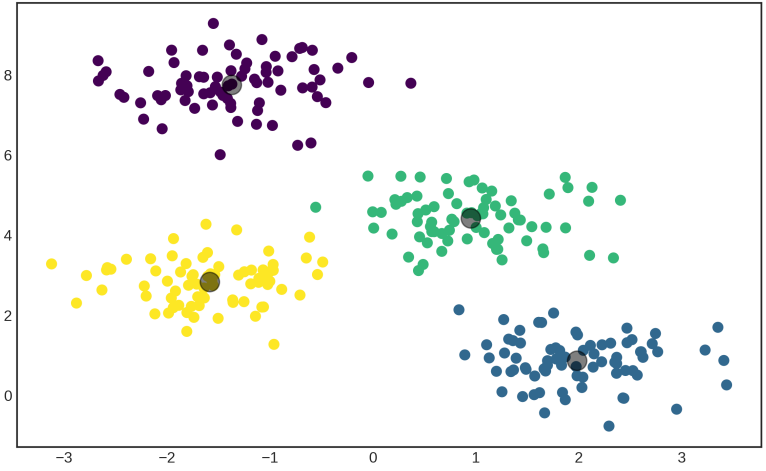
$$\sum_{i=0}^{k} \min_{\mu_j}(\|x_i - \mu_j\|^2)$$

- Each cluster is represented by the central vector or centroid μ_j.

K-means Clustering



K-means Clustering



4 clusters and their centroids

K-means Algorithm

Step 1: Randomly Assign Cluster Numbers

- Assign a number (1 to k) to each of the observations.
- This is the initial cluster assignment

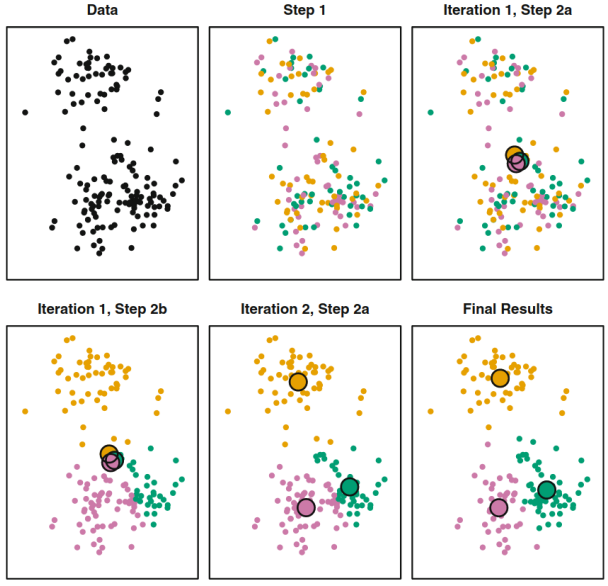
Step 2: Iterate Until Cluster Assignments Stop Changing

1. For each of the k clusters:
   - Compute the cluster centroid.
   - The k^{th} cluster centroid is the vector of the p feature means for the observations in the k^{th} cluster
2. Assign each observation to the cluster whose centroid is closest, where closest is defined using Euclidean distance.

Objective: Minimize Inertia

- The algorithm aims to choose centroids that minimize the inertia (within-cluster sum-of-squares criterion).

K-means Clustering

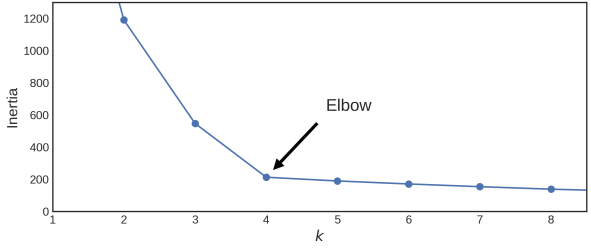


Finding the Optimal Number of Clusters

- Most of the time, the number of clusters does not stand out from looking at the data
- Inertia decreases with the number of clusters (e.g., each observation as a cluster)
- **Rule of Thumb: Choose the number of clusters at the "elbow"**

Finding the Optimal Number of Clusters

- Most of the time, the number of clusters does not stand out from looking at the data
- Inertia decreases with the number of clusters (e.g., each observation as a cluster)
- **Rule of Thumb: Choose the number of clusters at the "elbow"**



- The elbow is the point of inflection in the curve of inertia versus the number of clusters

Finding the Optimal Number of Clusters

- The **silhouette score** measures how well each point fits into its assigned cluster and how far it is from other clusters
- The silhouette score for a data point i is defined as:

$$\frac{b_i - a_i}{\max(a_i, b_i)}$$

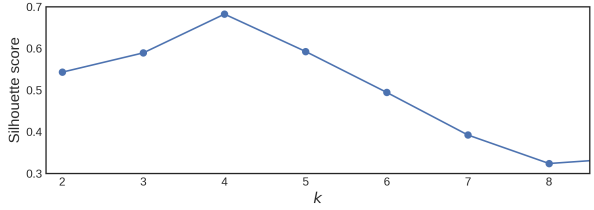
  where a_i is the mean distance between i and other points in the same cluster, and b_i is the mean distance between i and other points in the second closest cluster
- The optimal number of clusters can be selected based on the highest silhouette score.

Finding the Optimal Number of Clusters

- The **silhouette score** measures how well each point fits into its assigned cluster and how far it is from other clusters
- The silhouette score for a data point i is defined as:

$$\frac{b_i - a_i}{\max(a_i, b_i)}$$

  where a_i is the mean distance between i and other points in the same cluster, and b_i is the mean distance between i and other points in the second closest cluster
- The optimal number of clusters can be selected based on the highest silhouette score.



Latent Dirichlet Allocation (More on this next week...)

What is Latent Dirichlet Allocation (LDA)?

- LDA is a generative probabilistic model used for **topic modeling**
- It is widely used in natural language processing and has applications in text classification, sentiment analysis, and information retrieval
- It assumes that each **document** is a mixture of a few **topics** and that each topic is a probability distribution over **words**
- The model is used to infer the underlying topics that generate the observed documents

Latent Dirichlet Allocation (More on this next week...)

What is Latent Dirichlet Allocation (LDA)?

- LDA is a generative probabilistic model used for **topic modeling**
- It is widely used in natural language processing and has applications in text classification, sentiment analysis, and information retrieval
- It assumes that each **document** is a mixture of a few **topics** and that each topic is a probability distribution over **words**
- The model is used to infer the underlying topics that generate the observed documents
- LDA produces topics or themes that are **interpretable** and meaningful, whereas PCA and k-means produce clusters that may not have any inherent meaning or interpretation

Example

Media Slant is Contagious - Widmer, Galletta and Ash 2021

- 24 million article snippets from 600 U.S. local newspaper
- Identify topics from the text
- Use LDA by defining 128 topics
- Eventually, we have topic probabilities per newspaper

Topics from the newspaper-based LDA model

Table B.5: Newspaper-Based LDA Topic Model: List of 128 Topics

Most frequent tokens	Topic label	Local news
davi broker morgan stanley princeton	international economic actors	0
plant garden winter farmer flower	farmers	1
dairi payn lilli liabil utica	no label	0
probat fine suspend penalti ppg	crime	1
collin prayer omaha floyd billi	small city names	1
ice indiana chip hall fame	food	1
light marshal lane bennett home	local happenings	1
law immigr illeg enforc clinton	border control	0
harrison intellig counsel harper island	no label	0
park water lake land river	nature and infrastructure	1
springfield indian martinez tribe riley	local happenings	1
paul pope novel decatur roman	names	1
health care medic hospit center	healthcare	1
airport said nuclear plane iran	aviation and terrorism	0
oregon wine meter sullivan wildlif	nature and infrastructure	1
rate credit class flag glass	economy	0
secur report exchang act date	no label	0
offic agenc number address post	post	1
navi laker naval salina reagan	no label	0
walker nevada dear easter mother	family	1
bird boulder rent rumor wagner	farmers	1
weather snow temperatur day degr	weather	1

Example

CEO Behavior and Firm Performance - O. Bandiera, A. Prat, S. Hansen, R. Sadun

- They record diaries of 1,114 CEOs of manufacturing firms in six countries
- Use LDA to find the combination of features that best differentiate among CEOs
- They identify two CEO types
  - Manager: more time spent with employees
  - Leader: more time spent with C-suite executives
- Leader CEOs are more likely to lead more productive and profitable firms.

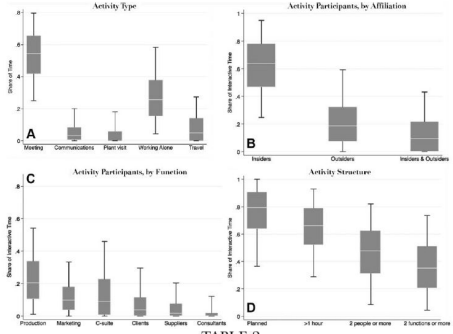


TABLE 2

MOST IMPORTANT BEHAVIORAL DISTINCTIONS IN CEO TIME-USE DATA

Feature	Times Less/More Likely
Less likely in behavior 1:	
Plant visits	.11
Just outsiders	.58
Production	.46
Suppliers	.32
More likely in behavior 1:	
Communications	1.90
Outsiders and insiders	1.90
C-suite	33.90
Multifunction	1.49