Unsupervised Learning

Computational Data Science,
Addis Ababa University

www.aau.edu.et

mesfin.diro@aau.eud.et

+251-912-086156

Unsupervised Learning

- When there is no label data, unsupervised learning techniques help in understanding the data by visualizing and compressing.
- The two commonly-used techniques in unsupervised learning are:
  - Clustering
  - Dimensionally Reduction
- Clustering helps in grouping all similar data points together.
- Dimensionality reduction helps in reducing the number of dimensions, so that we can visualize high-dimensional data to find any hidden patterns.

Clustering

- Clustering is a technique for finding similarity groups in data, called clusters.
- Clustering involves automatically discovering natural grouping in data.
- Unlike supervised learning (like predictive modeling), clustering algorithms only interpret the input data and find natural groups or clusters in feature space.
- The cluster may have a center (the centroid) that is a sample or a point feature space and may have a boundary or extent.
- Clustering can be helpful as a data analysis activity in order to learn more about the problem domain, so-called pattern discovery or knowledge discovery.

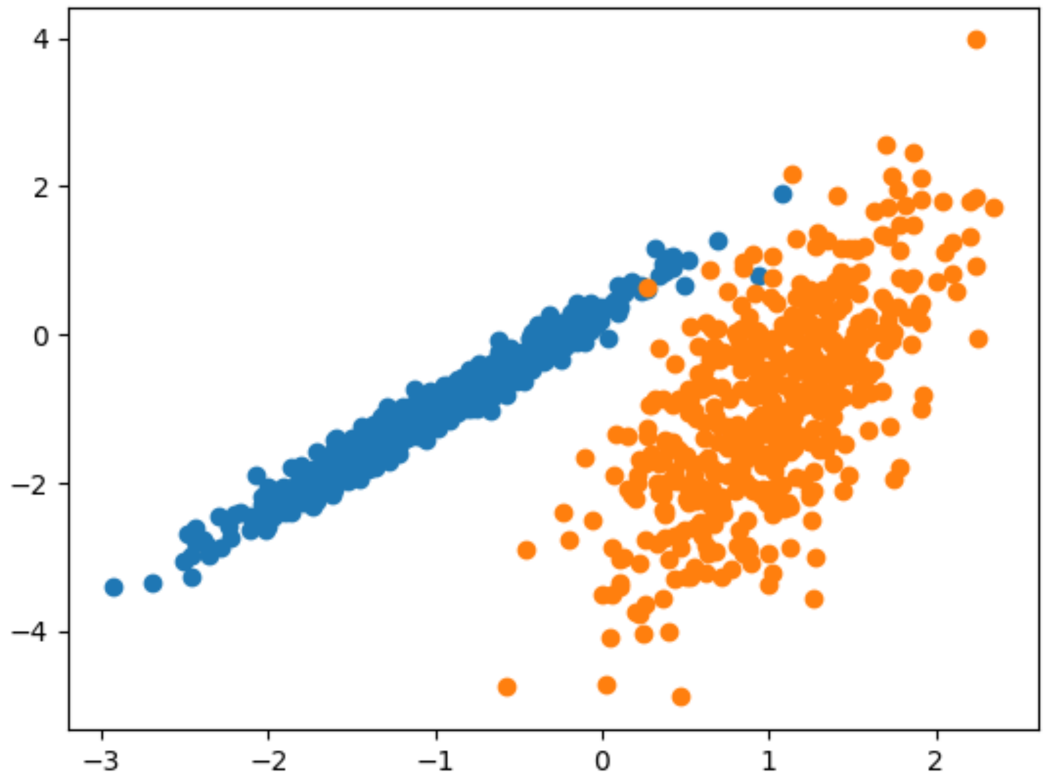
Aspects of clustering

- A clustering algorithm
  - Partitional clustering
  - Hierarchical clustering
- A distance (similarity or dissimilarity) function
- Clustering quality
  - inter-clusters distance maximized or
  - inter-clusters distance minimized
- The quality of a clustering result depends on the algorithm, the distance function, and the application

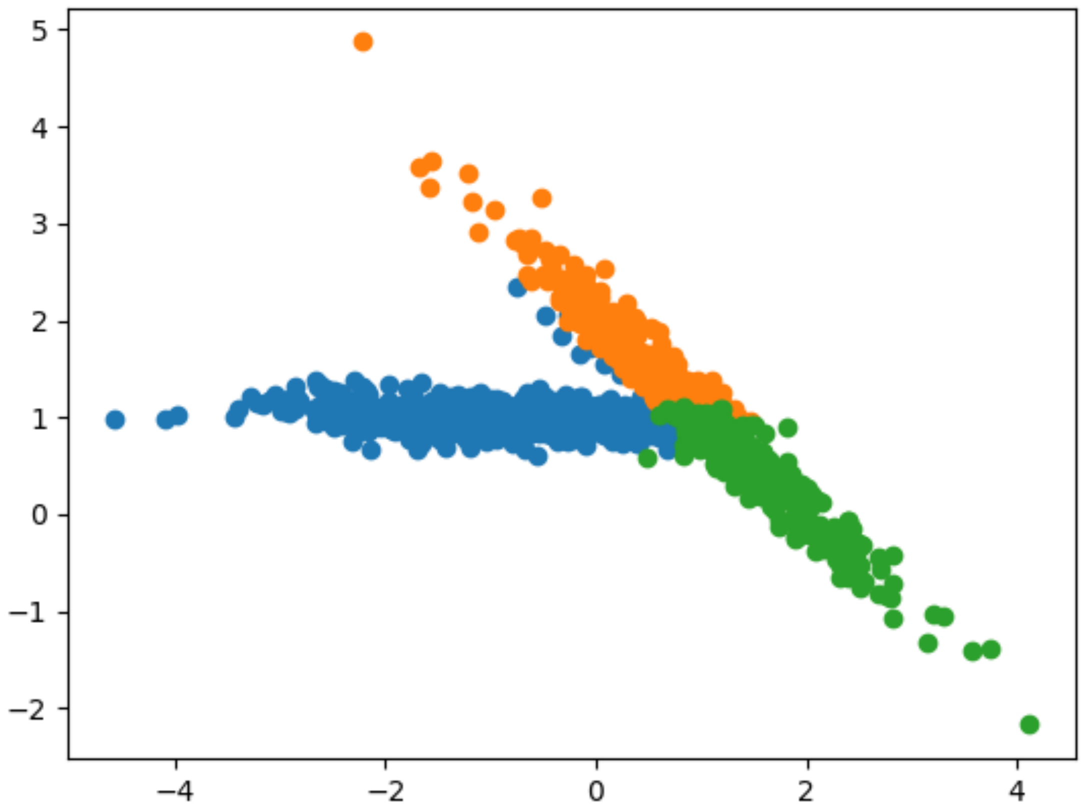
Clustering Algorithms

- There are many types of clustering algorithms.
  - K-means
  - Spectra Clustering
  - Mixture of Gaussian etc
- Many algorithms use similarity or distance measures between examples in the feature space in an effort to discover dense regions of observations.
- Central to all of the goals of cluster analysis is the notion of the degree of similarity (or dissimilarity) between the individual objects being clustered. A clustering method attempts to group the objects based on the definition of similarity supplied to it.

K-means Clustering Algorithms



K = 2



K = 3

K-means Clustering

- K-means is a partitional clustering algorithm
- The k-means algorithm partitions the given data into k clusters.
  - Each cluster has a cluster center, called centroid.
  - k is specified by the user
- The k-means algorithms can be used for any application dataset where the mean can be defined and computed.
- In the Euclidean, the mean of a cluster is computed with:

$$m_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

- where $|C_j|$ is the number of data points in the cluster C_j. The distance from one data points X_i to a mean(centroid) m_j is computed.

Weakness of K-means

- K-means algorithm is only applicable if the mean is defined.
  - For categorical data, k-mode - the centroid is represented by most frequent values.
- The user needs to specify k.
- The algorithm is sensitive to outliers
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Dimensionality Reduction

- Many modern data domains involve huge numbers of features / dimensions
  - Documents: thousands of words, millions of bigrams
  - Images: thousands to millions of pixels
  - Genomics: thousands of genes, millions of DNA polymorphisms
- Why reduce dimensions?
  - Redundant and irrelevant features degrade performance of some ML algorithms
  - Difficulty in interpretation and visualization
  - Computation may become infeasible
  - Curse of dimensionality

Approaches to Dimensionality Reduction

- Dimensionality reduction can be done in 2 ways:
  1. feature selection:keeping the most relevant variables
  2. features extraction: finding a smaller set of new variable
- Model regularization
  - L2 reduces *effective* dimensionality
  - L1 reduces *actual* dimensionality
- Combine (map) existing features into smaller
  - Linear combination(projection)
  - Non-Linear combination

Linear Dimensionality Reduction

- Linearly project *n*-dimensional data onto a *k*- dimensional space
  - $k < n$, often $k << n$
  - Example: project space of 10^4 words into 3 dimensions
- There are infinitely many *k*-dimensional subspaces we can project the data onto.
- Which one should we choose?

Linear Dimensionality Reduction

- Best *k*-dimensional subspace for projection depends on task
  - Classification: maximize separation among classes
    - Example: linear discriminant analysis (LDA)
  - Regression: maximize correlation between projected data and response variable
    - Example: partial least squares (PLS)
- Unsupervised: retain as much data variance as possible
  - Example: principal component analysis (PCA)

Linear Dimensionality Reduction

- **Best *k*-dimensional subspace for projection depends on task**
  - **Classification: maximize separation among classes**
    - **Example: linear discriminant analysis (LDA)**
  - **Regression: maximize correlation between projected data and response variable**
    - **Example: partial least squares (PLS)**
- **Unsupervised: retain as much data variance as possible**
  - **Example: principal component analysis (PCA)**

Variance and Covariance

- Variance is a measure of data spread in one dimension (feature)
- Covariance measures how two dimensions (features) vary with respect to each other
- $var(X) = \dfrac{\sum_{i=1}^{n} (X_i - \bar{X})((X_i - \bar{X})}{n-1}$
- $var(X, Y) = \dfrac{\sum_{i=1}^{n} (X_i - \bar{X})((Y_i - \bar{Y})}{n-1}$

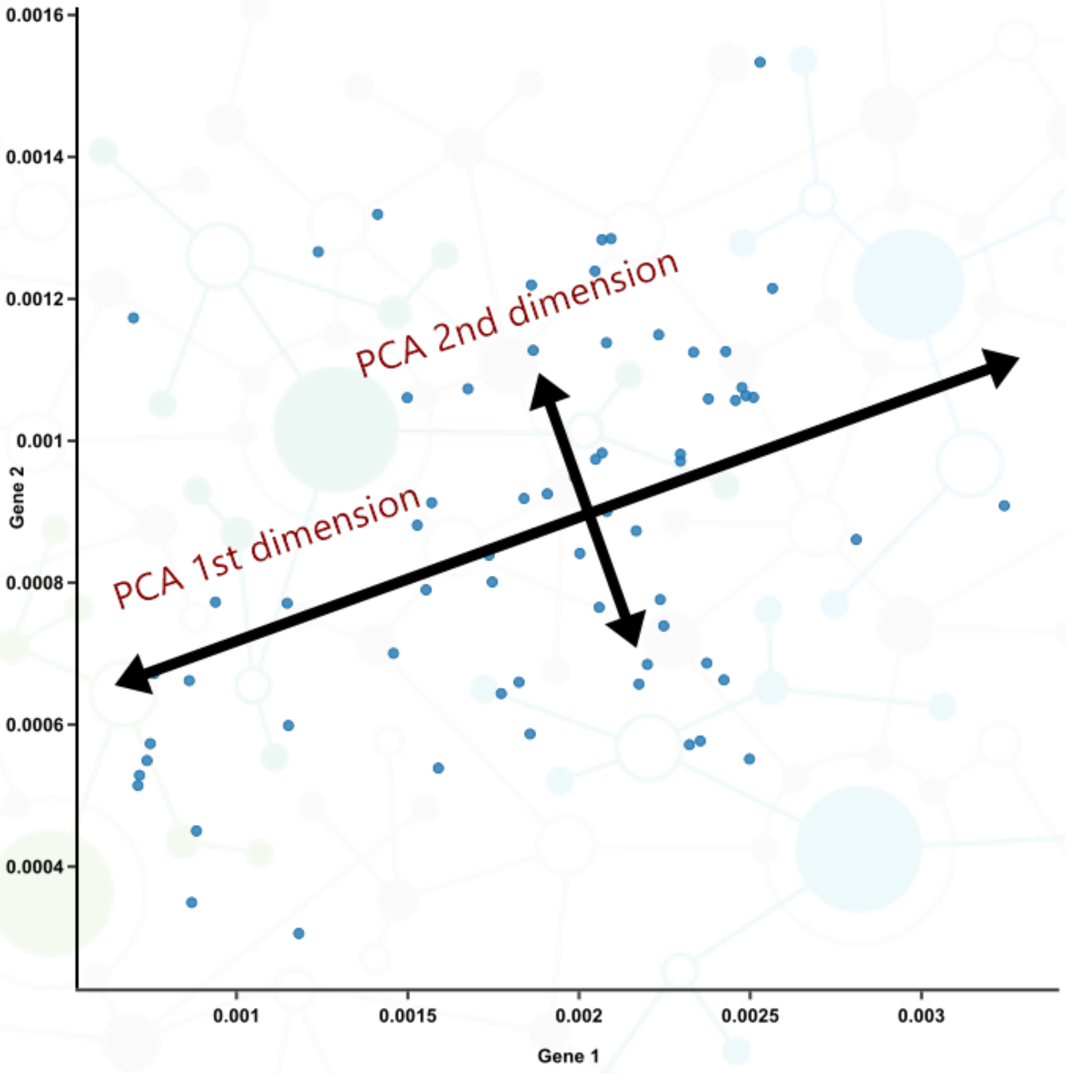
Variance and Covariance Matrix

- Considering the sign (rather than exact value) of covariance:
  - Positive value means that as one feature increases or decreases the other does also (positively correlated)
  - Negative value means that as one feature increases the other decreases and vice versa (negatively correlated)
  - A value close to zero means the features are independent
  - If highly covariant, are both features necessary?
- Covariance matrix is an $n \times n$ matrix containing the covariance values for all pairs of features in a data set with n features (dimensions)
- The diagonal contains the covariance of a feature with itself which is the variance (which is the square of the standard deviation)
- The matrix is symmetric

Principal Component Analysis (PCA)

- Widely used method for unsupervised, linear dimensionality reduction
- GOAL: account for variance of data in as few dimensions as possible (using linear projection)
- First principal component is the projection direction that maximizes the variance of the projected data
- Second principal component is the projection direction that is orthogonal to the first PC and maximizes variance of the projected data
- Find a line, such that when the data is projected onto that line, it has the maximum variance.
- Repeat until have *k* orthogonal lines

Principal Component Analysis (PCA)



Steps in PCA

- Mean center the data D
- Compute covariance matrix of data D
- Calculate eigenvalues and eigenvectors of covariance matrix
  - Eigenvector with largest eigenvalue λ_1 is the 1^{st} principal component
  - Eigenvector with k^{th} largest eigenvalue λ_k is the k^{th} PC
  - $\frac{\lambda_k}{\sum \lambda_i}$ is the proportion captured by k^{th} PC
- Rank the eigenvalues in decreasing order
- Select the eigenvalues that retain fixed percentage of variance
  - E.g ($80\,\%$ the smallest d such that $\frac{\sum_i^d \lambda_i}{\sum_i \lambda_i} \geq 80\,\%$)

THANK YOU!