Identifying Cluster Structures in High-dimensional Data

Anjoreoluwa Boluwajoko, Abdalah Namwenje, Mohamed Hussien, Gisel Kuatse

Department of Computer Science and Applied Mathematics
Wits University

January 25, 2025

Overview

Problem Statement

The purpose of the presentation is to apply unsupervised machine learning techniques to the high-dimensional data in order to obtain an optimal cluster structure.

The Importance and Applications of Clustering High dimensional Data

Feature	Description
Importance	Dimensionality reduction, noise reduction,pattern discovery,anomaly-detection.
Applications	Bioinformatics, image processing,market segmentation,Finance.

Table: Summary of the importance and applications of clustering high-dimensional data.

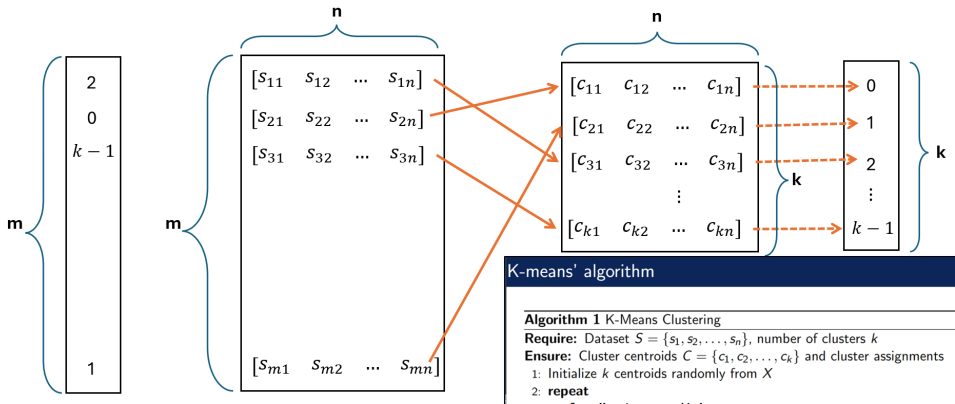
Presentation of Data

K-mean Clustering

Definition

K-means clustering is an unsupervised machine learning algorithm used to partition data into k distinct clusters. The goal of the algorithm is to group data points with similar features in a unique cluster.

K-means Clustering



$\boldsymbol{n}$ – *dimension*

$\boldsymbol{m}$ – *number of data points*

$\boldsymbol{k}$ – *number of clusters*

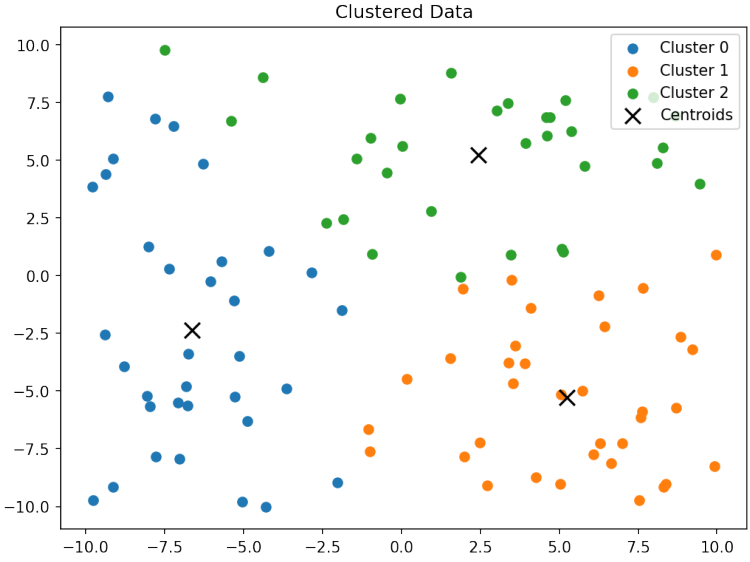
K-means' algorithm

Algorithm 1 K-Means Clustering

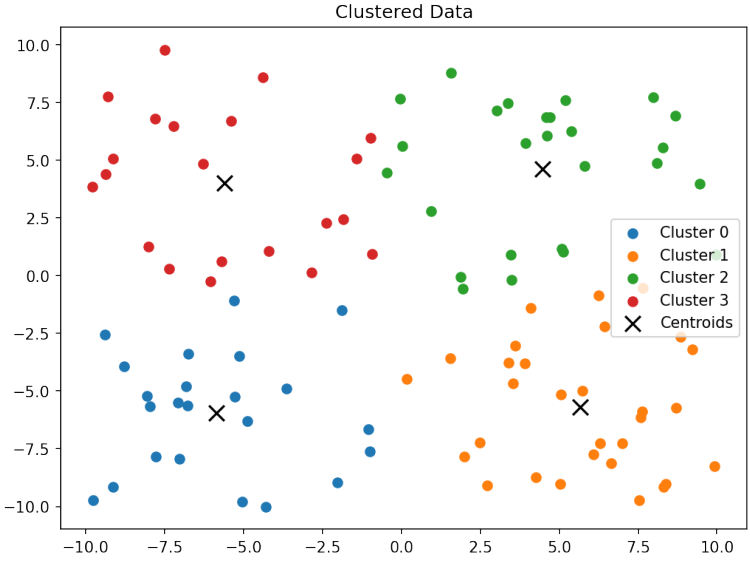
Require: Dataset $S = \{s_1, s_2, \dots, s_n\}$, number of clusters k
Ensure: Cluster centroids $C = \{c_1, c_2, \dots, c_k\}$ and cluster assignments

1: Initialize k centroids randomly from X
2: **repeat**
3: **for all** points $x_i \in X$ **do**
4: Assign x_i to the nearest centroid c_j using Euclidean distance
5: **end for**
6: **for all** centroids c_j **do**
7: Update c_j to be the mean of all points assigned to it
8: **end for**
9: **until** centroids do not change or change is below a threshold
10: **return** final centroids C and cluster assignments

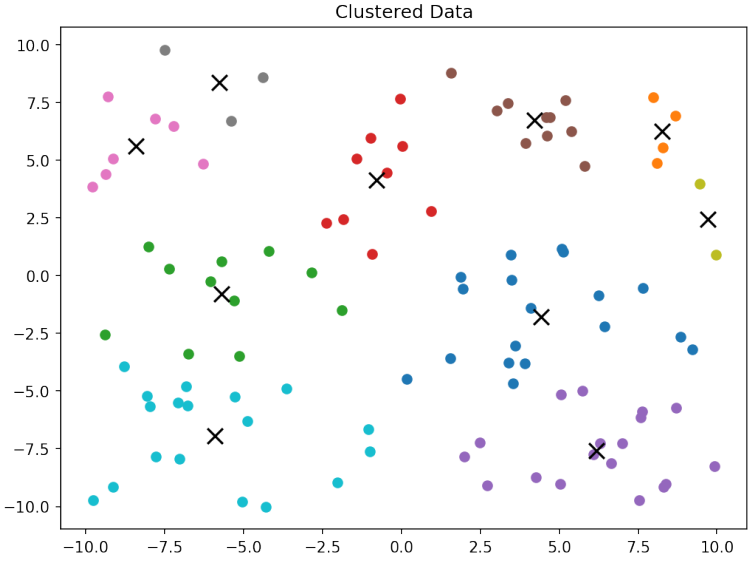
K-means Clustering, k = 3



K-means Clustering, k = 4



K-means Clustering, k = 10



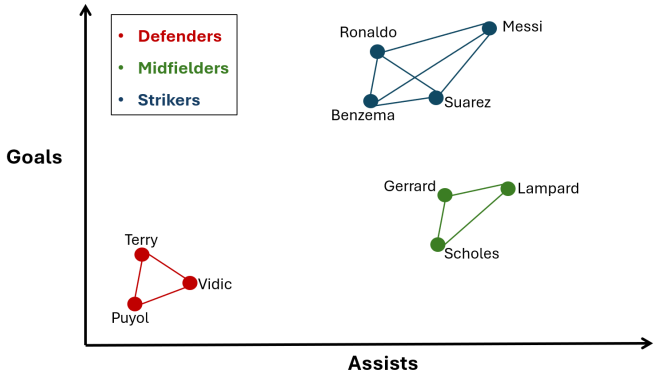
Limitations of K-means Clustering

Issue	**Impact**	**Possible Improvement**
Fixed number of K-cluster	Incorrect choice of K can lead to over clustering or under clustering	Use Elbow Method
Dimensionality	Distance becomes less meaningful in high dimensions, hence reducing clustering quality	Apply dimensionality
Sensitive to outliers	Outliers can shift cluster centriods, leading to inacurate cluster assignments	Detect and remove outliers using K-medioids

Graph Theory for Dummies

Definition

Data can be treated as a graph. A **graph** G consists of vertices and edges where vertices represent the data points and edges represent the similarity/connection between the data point.



Spectral Clustering

Definition

Spectral clustering is an unsupervised machine learning algorithm that uses graph theory to partition data into clusters by representing the data points as a graph and using eigenvalues of a similarity matrix from the graph to find clusters.

Figure: Spectral Clustering General Algorithm

Spectral Clustering General Algorithm

•Represent the data point as a complete graph where the data points are the vertices and the edges are the similarities

•Assign a weight to each edge using the formula

$$a_{ij} = \exp\left(-\frac{\left|\left|s_i - s_j\right|\right|_2^2}{2\,\sigma^2}\right)$$

•This is the **Adjency Matrix,** $A = (a_{ij})$

Spectral Clustering General Algorithm

Construct the Graph Laplacian:

- Construct a **Diagonal Matrix D** such that

$$D_{ii} = \sum_{j=1}^{n} a_{ij}$$

- Unnormalized **Laplacian** Matrix:

$$L = D - A$$

- Normalized **Laplacian** :

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

Spectral Clustering General Algorithm

- **Eigenvalues:** They are all non-negative because the matrix L is symmetric positive definite.
- Select the first k smallest eigen values $\lambda_i \neq 0$.
- **Eigenvectors:** Form the new feature subspace, matrix U, by taking the eigenvectors u_i that correspond to the chosen eigenvalues.
- The columns of U are the calculated eigenvectors.

$$U = [u_1, \quad u_2, \quad \cdots, \quad u_k]$$

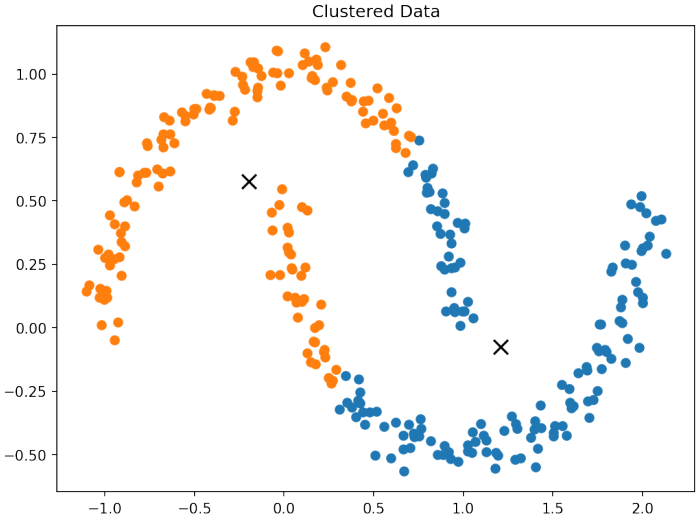
Spectral Clustering General Algorithm

- Treat the rows of U as data points.
- Create U_{norm} by normalizing the rows of U
- Apply K – means to the rows of U

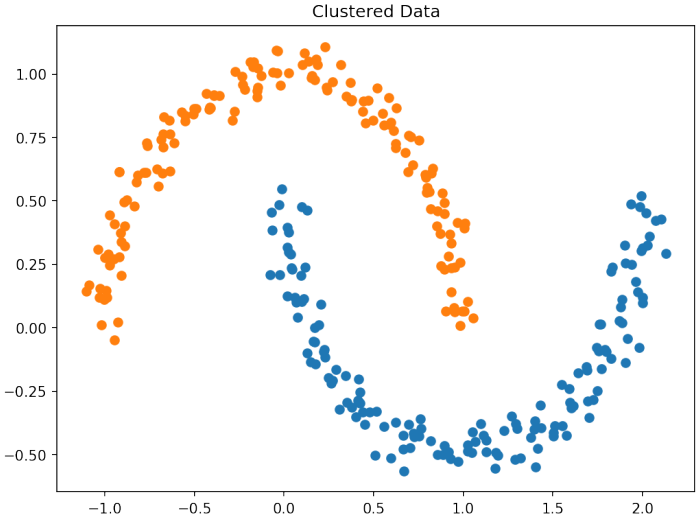
Spectral Clustering General Algorithm

- Assign the labels of the data points in U_{norm} to the original data points in S

Moons Clustering With k-means Algorithm



Moons Clustering With Spectral Clustering Algorithm



Spectral Clustering Data

Limitations of Spectral Clustering

Issue	**Impact**	**Possible Improvement**
High computational cost ($O(n^3)$)	Slow for large datasets	Approximation methods (e.g., Nyström) for eigenvalues
Parameter sensitivity(σ)	Poor clustering with bad parameters	Use heuristics for k, median for all pairwise distances
Graph connectivity issues(Assuming all points are connected)	Incorrect eigenvector calculations and clustering	Using compressed sensing theory

Compressed Sensing Theory

- The more zeros a matrix has, the easier it becomes to compute the eigenvalues and this gives us fewer nonzero eigenvalues to deal with.

Theorem: Compressed Sensing Theorem

Assuming that a dataset is:

i. Self-expressive

ii. Noise free

iii. Has clusters that are independent and disjoint

Compressed sensing is a signal processing technique for efficiently acquiring and reconstructing a signal, by finding solutions to undetermined linear systems.

Sparse Optimization

- According to Compressed Sensing Theorem, data points in the same cluster can be represented as linear combinations of each other.
- Sparse Optimization Helps create a matrix A with as few non-zero elements as possible using the constrained objective function:

$$\min \quad \|w_i\|_1 \quad s.t. \quad s_i = Sw_i, \qquad w_{ii} = 0 \tag{1}$$

- The constraint w_{ii} eliminates the trivial solution of writing a point as a linear combination of itself.
- The system is undetermined hence there are infinetly many solutions. The main idea is that amoung all solutions, there exists a sparse solution, w_i, whose nonzero entries correspond to data points from the same subspace as s_i.
- After solving for the W matrix, the Adjency Matrix can be computed as:

$$A = |W| + |W|^T$$

Objective function

- The constrained objective function can be compactly written in matrix form as

$$\min \|\mathbf{W}\|_1 \qquad \text{s.t.} \qquad \mathbf{S} = \mathbf{SW} \quad , \quad \text{diag}\,(\mathbf{W}) = \mathbf{0} \tag{2}$$

- The unconstrained objective function takes the form

$$\min_W F(W) = \mu \|W\|_1 + \frac{1}{2}\|SW - S\|_f^2 \quad , \quad diag(W) = 0 \tag{3}$$

- Solving this problem is a mess because for some input matrix B

$$\|B\|_f^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^2 \rightarrow \text{smooth} \qquad \textit{and} \qquad \|B\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |b_{ij}| \rightarrow \text{nonsmooth}$$

FISTA

Fast Iterative Shrinkage-Threshold Algorithm

Algorithm 1 FISTA with Matrix Input

Initialize $Z_1 = W_0 = 0 \in \mathbb{R}^{n \times n}$, $t_1 = 1$
for $k \geq 1$ **do**:
    $W_k = p_{\mu\alpha}(Z_k)$ (hold your questions!)
    $diag(W_k) = 0$
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
    $Z_{k+1} = W_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(W_k - W_{k-1})$
        break if $\|W_{k+1} - W_k\|_F < \text{tol}$
        else $W_{k+1} = p_{\mu\alpha}(Z_{k+1})$
    return W_k

Shrinkage Operator

let $x \in \mathbb{R}^n$ For some function

$$F(x) = f(x) + g(x)$$

where $f(X)$ is a smooth function, $g(X)$ in non smooth. We can use the quadratic approximation of F at a given point $y \in \mathbb{R}^n$ as

$$Q_\alpha(x, y) \approx f(y) + \nabla f(y)^T (x - y) + \frac{1}{2\alpha}\|x - y\|_2^2 + g(x)$$

if $g(x) = \mu\|x\|_1$, $Q_{\alpha\mu}(x, y)$ admits a unique minimizer

$$p_{\mu\alpha} = arg\min_x \{Q_{\alpha\mu}(x, y) : x \in \mathbb{R}^n\}$$

Shrinkage Operator

$$p_{\mu\alpha} = arg\min_{x}\{Q_{\alpha\mu}(x, y) : x \in \mathbb{R}^n\}$$

The solution to this is called the shrinkage operator where for some input $v \in \mathbb{R}$

$$\mathbb{T}_{\mu\alpha}(v) := \max\{0, |v| - \mu\alpha\} \cdot sgn(v)$$

References

- R. Xu and D. C. Wunsch II, "Survey of clustering algorithms," IEEE Trans. Neural Networks, vol. 16, no. 3, pp. 645–678, 2005.
- K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbour" meaningful?" in Int. Conf. Database Theory. Springer, 1999, pp. 217–235.
- L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 90–105, 2004.
- R. Vidal, "Subspace clustering," IEEE Signal Processing Magazine, vol. 28, no. 2, pp. 52–68, 2011.
- Beck, A. and Teboulle, M., 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences, 2(1), pp.183-202.

The End