k-Plane Clustering

Introduction

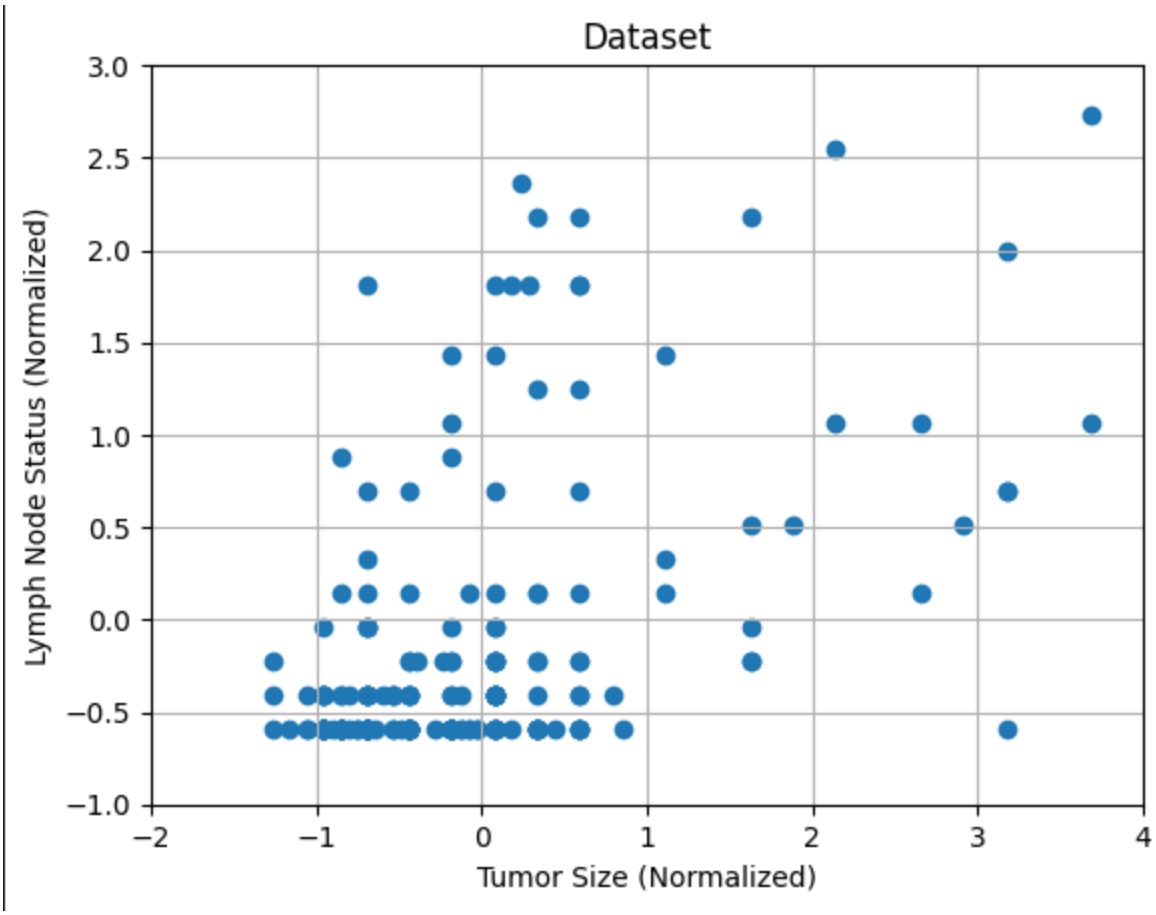
- Research Paper : k-Plane Clustering
- Authors : Bradley and Mangasarian
- Paper Link : https://doi.org/10.1023/A:1008324625522

Goal

- Our goal is to cluster `m` given points in `n`-dimensional real space into `k` clusters by generating `k` hyperplanes.
- We iteratively repeat cluster assignment and cluster update until the algorithm finally converges and we get the minimum sum of squares of distances of each point to its nearest plane.

Dataset

- The algorithm is implemented for the Wisconsin Prognostic Breast Cancer (WPBC) Dataset. It can be found at https://archive.ics.uci.edu/dataset/16/breast+cancer+wisconsin+prognostic
- The features we use are `Tumor Size` and `Lymph Node Status` , both normalised to have `0` mean and `1` standard deviation.
- We have a total of `198` points in the dataset and we will be dividing them into `3` clusters.



Notation

- `m` - total number of data points (198)
- `n` - dimensions (2)
- `k` - number of clusters (3)
- `A` - collection of all datapoints in a `m x n` matrix
- `w` - collection of all weights (norm = 1) in a `k x n` matrix
- `b` - collection of all bias terms in a `k` length array
- `m_cluster` - number of datapoints belonging to a particular cluster
- `A_cluster` - collection of all datapoints belonging to a particular cluster in a `m_cluster x n` matrix

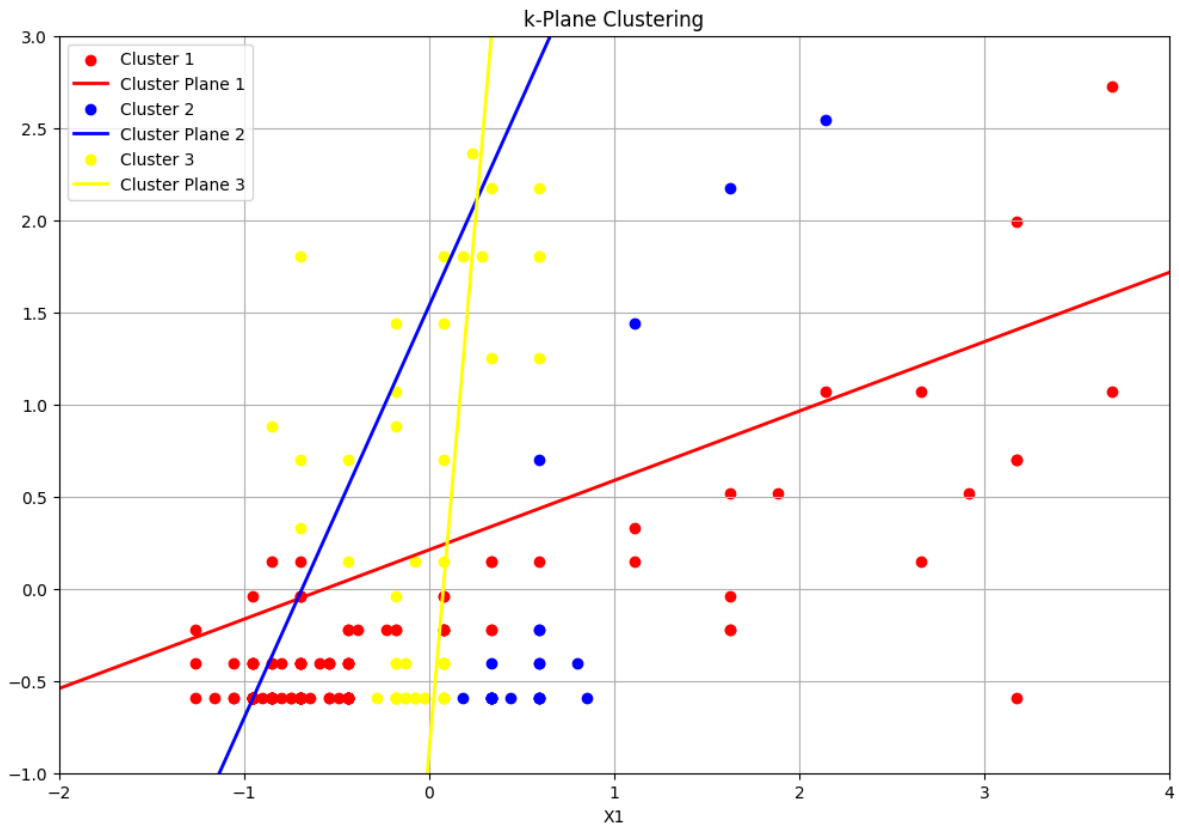
Algorithm

- **Cluster Assignment**
  - assigning points to the nearest cluster plane
  - for each point, our goal is to determine the index of plane closest to it
  - $|A_i w_{l(i)}^j - \gamma_{l(i)}^j| = min_{l=1,2...k} |A_i w_l^j - \gamma_l^j|$,

    where $l(i)$ is the index of closest plane for A_i
- **Cluster Update**
  - for each cluster, we collect all its datapoints and try to find the plane which minimises the sum of squares of distances of each point to itself
  - $B(l) = A(l)' * (I - \frac{ee'}{m(l)}) * A(l)$,

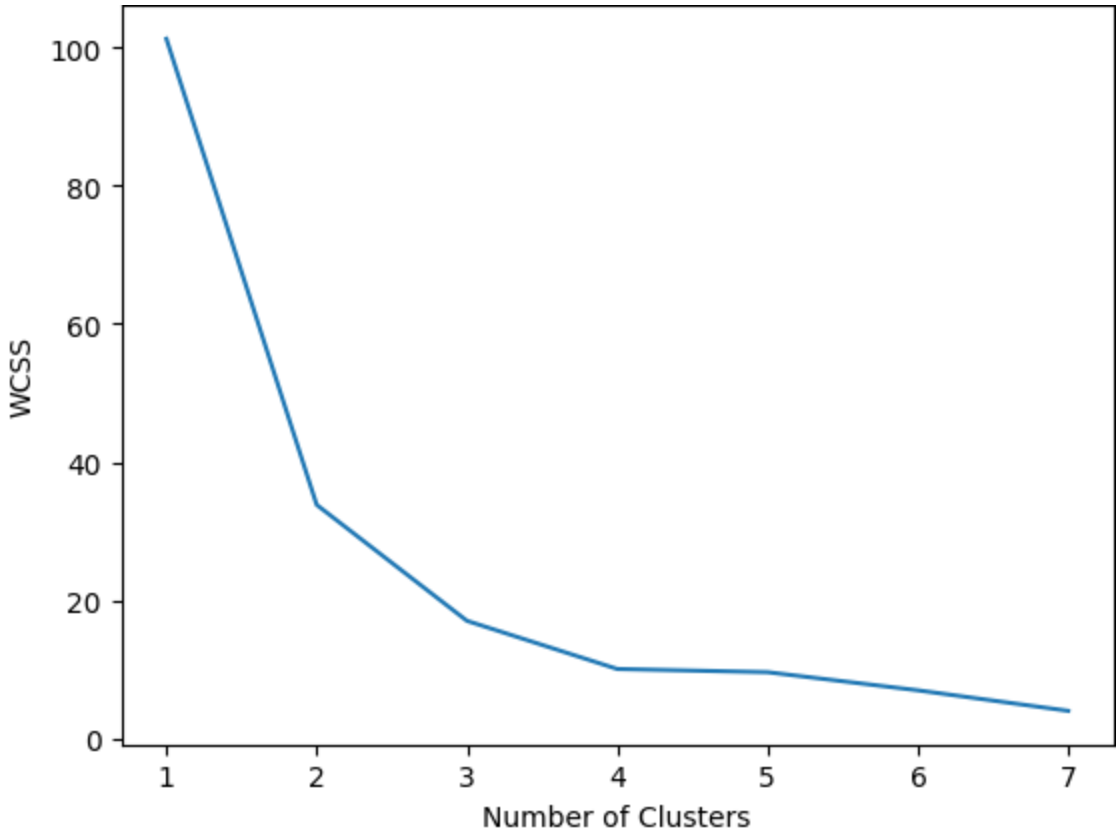
    where $A(l)$ is the collection of all datapoints in cluster l and e is a vector of ones of appropriate dimension
  - Then, corresponding to the smallest eigenvalue for B, we find the eigenvector and set the value of w as

    $w_l^{j+1} = v$ (v = eigenvector corresponding to smallest eigenvalue)
  - $\gamma_l^{j+1} = \frac{e' * A(l) * w_l^{j+1}}{m(l)}$
- **Finite Termination**
  - The kPC algorithm terminates in finite step at a locally optimal cluster assignment.



Extra

- Similar to k-means, we can use the “elbow method” to find the ideal number of clusters to use for our dataset


100
80
60
40
20
0
WCSS
1
2
3
4
5
6
7
Number of Clusters