Week 11 :-

Unsupervised learning - clustering

PAGE No. / DATE / /

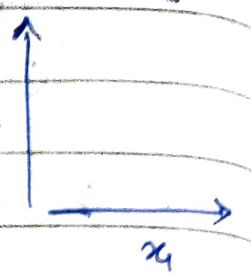
Objectives :

- Applications
- Understand objectives & algorithms
- Methods to assess goodness

Main objective :

Sample points without any target values. - Given

Find natural subsets of data samples.



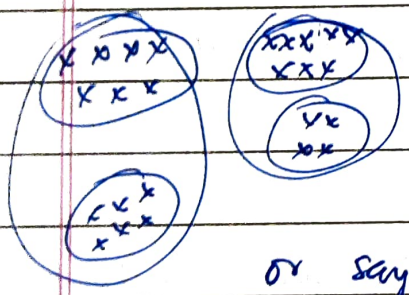
Applications :

- Natural groups
- Reduce data and deal with only one
- find outliers

Modelling joint probability density - Multimodal (Each mode corresponding to cluster)

Clustering is hard because it is unsupervised :

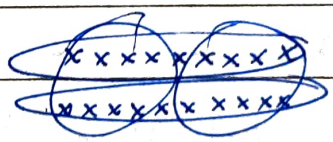
Eg.



N samples $\Rightarrow$ 1 - N clusters

we don't know which is good.

or say :



What clustering is better?

Lots of subjectivity.

Hard partitioning using K means clustering :

- Minimize sum of squared distances from cluster center.

Hard partioning $\Rightarrow$

If $x_i \in C_k \Rightarrow x_i \notin C_j \quad \forall \quad j \neq k$

S_k is set of points belonging to C_k

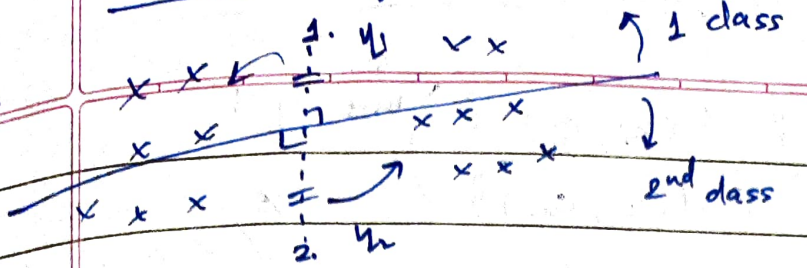
Minimize $$\sum_{x_i} \sum_{k} \mathbb{1}_{x_i \in C_k} \| x_i - y_k \|^2$$

$$y_k = \frac{1}{|C_k|} \sum_{x_i} \mathbb{1}_{x_i \in C_k} \cdot x_i$$

K-means, single iteration:

PAGE No. / DATE / /

Eg:



K = 2 (Suppose)

Randomly initialize centroid positions:

Iteration loop until Δ centroid location $< \epsilon$

1. Compute membership of each point
   $(\text{Arg min}_j \, d(x_i, y_j))$
2. Recompute y_j for all j

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

How to initialize centroids? :

[This is entire Kmeans algorithm.]

- Random initialization
- Take point, pick furthest point from that, then from other, so on.

Complete K means:

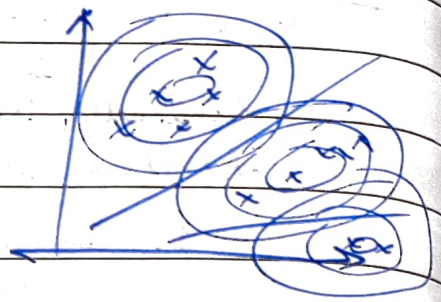
- Stopping criteria:
  - Whenever centroid computed $\because$ membership depends on d_{min} $\Rightarrow$ Distance can never increase.
    $\Rightarrow$ Each iteration at least better than previous.
  - Also, each point can be in one of K classes
    $\Rightarrow$ K^N total memberships $\Rightarrow$ Finite iterations
    $\Rightarrow$ Average distance can be observed in finite time
  - K means NOT guaranteed to find minimum $\sum \| - \|^2$ square distance sum.

FAQs: How to choose K? How to do soft clustering?

10-2 (Fuzzy C-Means; DBSCAN)

PAGE No.
DATE / /

$$w_{ij} = \left[\sum_k \left(\frac{\{ d(x_i, c_j) \}}{\{ d(x_i, c_k) \}} \right)^{\frac{2}{m-1}} \right]^{-1}$$



Weights are w_{ij}^m

layer m → More squad out clustering

$m \geq 1$

Sum of classes weights = 1

Here we minimized weighted sum of square differnce

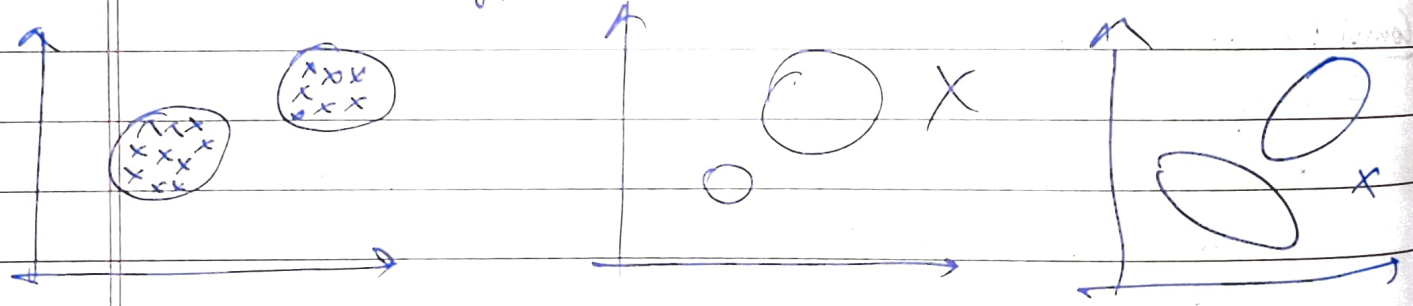
m - Controls fuzziness factor ($m = 1 \Rightarrow$ k mean hard clustering)

$m = 1 \Rightarrow L^\infty$ norm (Minimum only considered)

$m > 1 \Rightarrow$ More fuzziness

Annealing: Start with laye m, keep reducing to get harder clustering.

Prior: Equal sized hyperspheres (isotropic) [Drawback]



(Why? Because we are taking Euclidean distance Clustering)

DBSCAN: Density Based Scanning:

- High density cores of arbitrary shapes
- Medium density peripheries
- lower density outlier regions

DBSCAN Criteria :-

PAGE No. / DATE / /

- Hyperparameters
  - Min points m | - Tolerance ϵ
- Three type of points
  - Core : m points in ϵ radius
  - Reachable : Not core, but $\exists$ path from core with hops $< \epsilon$
  - Outliers : Others

Increasing m $\Rightarrow$ More stringent criteria, fewer core points
Increasing ϵ $\Rightarrow$ looser, merging of clusters

How to choose m & ϵ ? Data visualization.

Algorithm: (Crude)

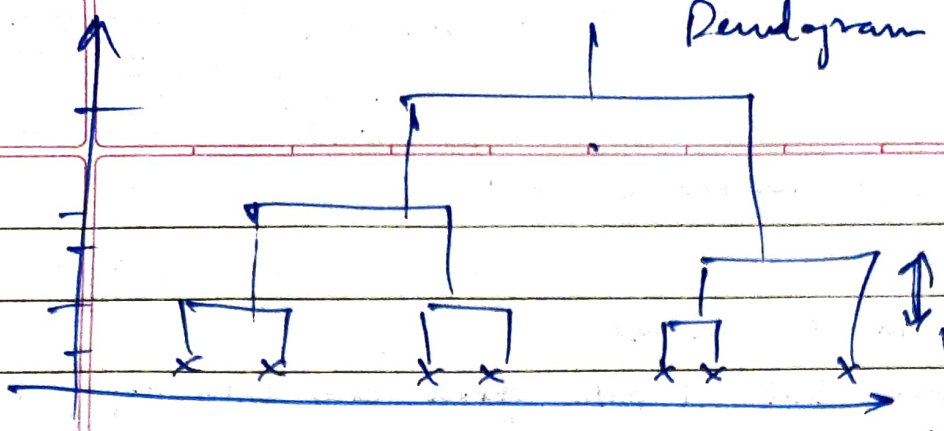
For each sample x_i
  for each other
    Mark j as neighbor of i if $d_{ij} < \epsilon$
    Increment neighbours n_i of x_i

} Adjacency discovery

for each sample x_i
  Neighbors $\geq m$ $\Rightarrow$ Core
  Neighbor > 0 $\Rightarrow$ Reachable
  $= 0$ $\Rightarrow$ Outlier

For each sample x_i
  Core & unclustered, then mark all connected samples with this cluster. $\rightarrow$ ∵ They are reachable
  (DFS or BFS and mark all as same cluster)

★ Hierarchical clustering gives all possible clusters as dendograms
- Start each as singleton - Merge iteration. (Closest clusters merged)



Within cluster ⇒ Variance of data ~ = Variance of 1 cluster

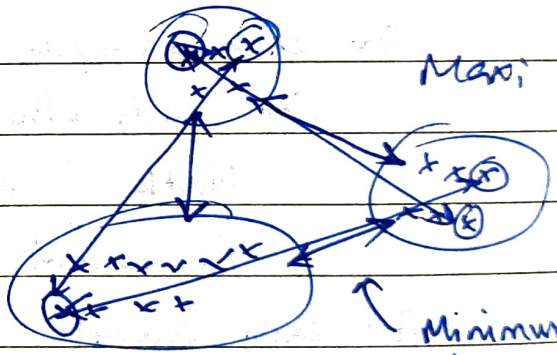
(Criteria) Variance = 0 for N point clusters.

(10-3) Clustering due diligence: [20:58]

Some types of clusters distances

- Min: Single-linkage
  Min dist. between 2 clusters
- Max: Complete-linkage
  Max dist. bw 2 clusters
- Avrge:
  Distance bw centroids

Eg:



Which to merge? ⇒ Minimum.

Maximum linkage: Take max distance bw clusters. The minim of those is merged.

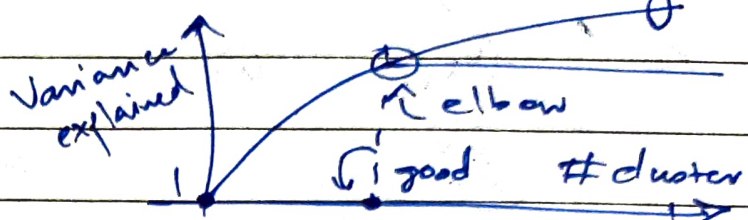
Avrge: Centroids closest to each other are merged.

Order of merging changes based on type of min distance.

Kind of distance may also be different:

PAGE No. / DATE / /

- Euclidean
- Mahalanobis dist
- Some dissilimarity.



Clustering Criteria:-

Several Methods:

- Dunn's Validity Idx
- Silhouette method
- C-index
- Goodman-Kruskal idx
- Elbow method
- Davies Bouldin idx

(1) **Elbow method**: Variation vs. no. of clusters

- Between group variance & total variance

Total variance - Average or max within cluster variance

[Did not understand.]

[Total variance - Averge or max Var among clusters.]

(2) **Davies Bouldin idx**:

Minimize ratio of intra-cluster versus inter-cluster var

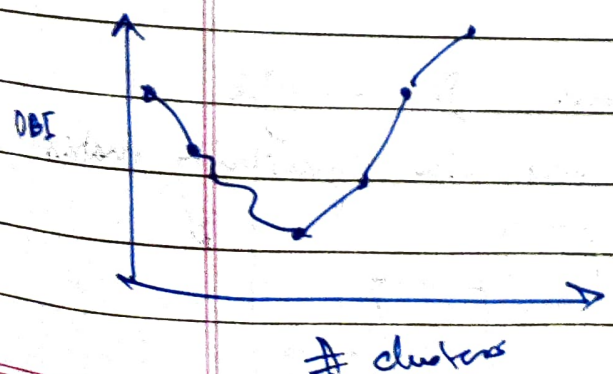
⇒ We want min. radius WITHIN cluster

⇒ We want more dist. between clusters

S = Variance

So, $\underset{i}{\text{Avg}} \; \underset{j}{\text{Max}} \left[\frac{S_i + S_j}{M_{ij}} \right]$ ← clusters i & j

M_{ij} ← variance for all points in cluster or dist bw centroids

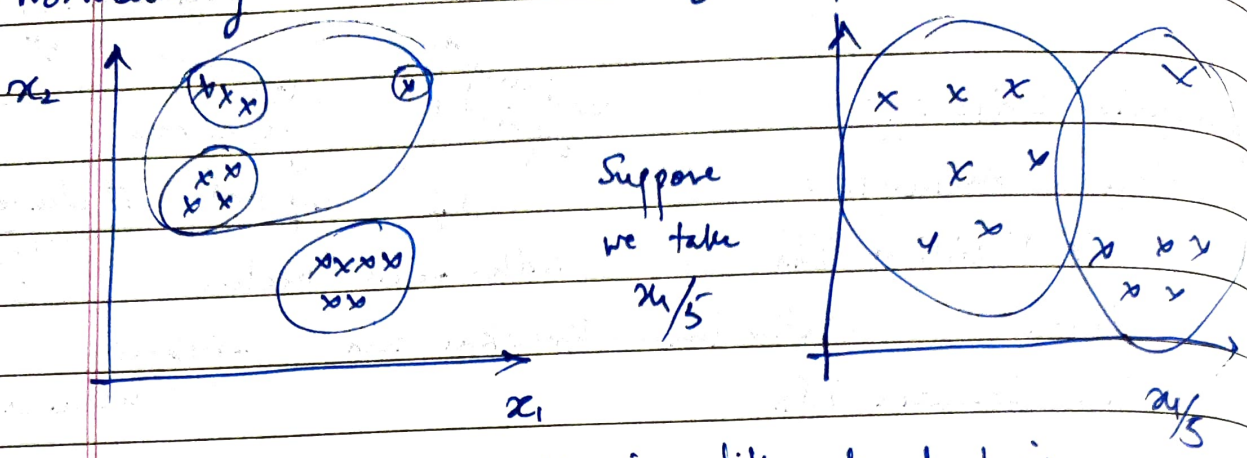


May be noisy, however

Impact of variable transforms on clustering :-

PAGE No.
DATE / /

- Normalizing variables
- Log or power transforms

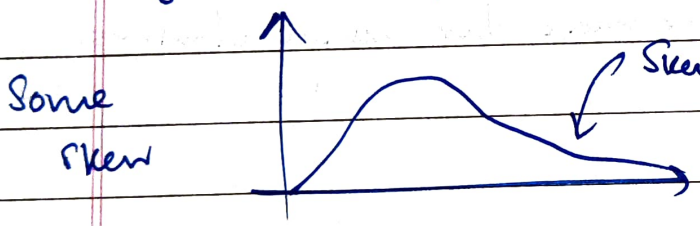


Changing an axis may result in different clustering.

⇒ Should normalize variance along axis

⇒ Should look at if some variables may need to be converted to log or power transform.

Eg. Revenue per customer :



[Violates assumption of isotropy in directions]

Log transform may compress higher values ⇒ More Gaussian.

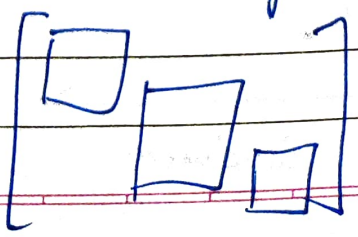
Advanced topics :

- Gaussian mixture models using EM algorithm
- Spectral clustering
  - $x_1, x_2 \dots x_N$ of dimensions $g_1 \dots g_d$

Similar to each other may be seen via similarity matrix

$N \times N$ matrix

i-j sample similarity



Permute rows & columns such that the values may be high in blocks

PAGE No.
DATE / /

Similarity pattern high when values high.

⇒ Eigen analysis done on similarity matrix
Spectral clustering metrics (Graph cuts etc)