



CS-F415: Data Mining

15

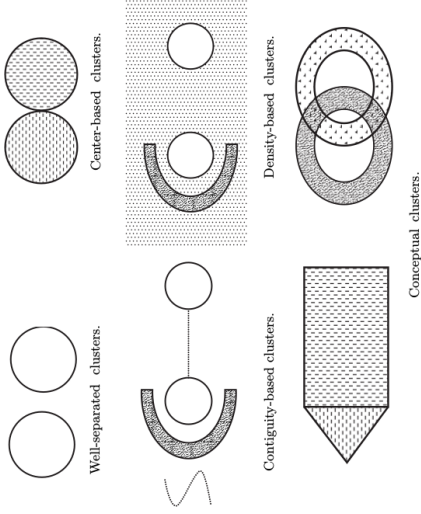
Clustering K-Means



Dr. Kamlesh Tiwari
Associate Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA
March 28, 2024 **MW/F 4:00pm** 6:101 @ BITS-Pilani [Jan-May 2024]

<http://ktiwari.in/dm>

Clustering Approaches



K-means Algorithm

Number of clusters *i.e.* the value of *K* is provided by the user

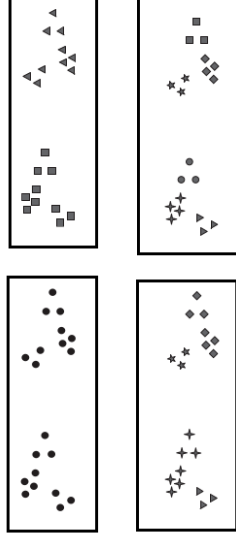
Algorithm 1: K-means

- 1 Randomly select *K* points as centroids
- 2 **repeat**
- 3 **foreach** datum point *d_i*, **do**
- 4 Assign *d_i* to one of the closest centroids (thereby forming *K* clusters)
- 5 Recompute centroid (mean) for each cluster
- 6 **until** *The centroids converge*;

Closeness is measured by **Euclidean distance**, cosine similarity, correlation, Bregman divergence *etc*

Clustering

Grouping data - based on their **homogeneity** (similarity or closeness).



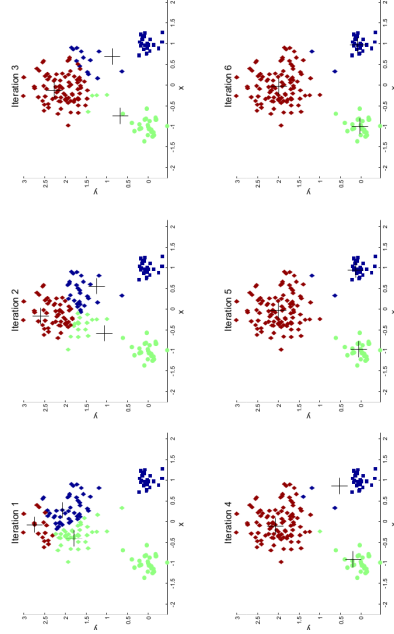
Objects within a group are similar (or related) and are different from the objects in other groups. **When is it better?**

Clustering

- **Unsupervised** in nature (i.e. right answers are not known)
- Clustering is useful to 1) Summarization, 2) Compression, and 3) Efficiently Finding Nearest Neighbors
- **Type:**
 - ▶ Hierarchical (nested) versus Partitional
 - ▶ Exclusive versus Overlapping versus Fuzzy
 - ▶ Complete versus Partial
- **K-means:** This is a prototype-based¹, partitional clustering technique that attempts to find a user-specified number of clusters (*K*), which are represented by their centroids.

¹ object is closer (more similar) to a prototype

K-means in Action



Evaluation of K-means²

For a given data set $\{x_1, x_2, \dots, x_n\}$, let K-means partitions it in $\{S_1, S_2, \dots, S_K\}$ then the objective is to minimize

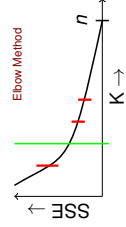
$$\operatorname{argmin}_S \sum_{i=1}^K \sum_{x \in S_i} \operatorname{dist}^2(x, \mu_i)$$

where μ_i corresponds to i^{th} centroid. $\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x$

- Typical choice for dist function is Euclidean Distance

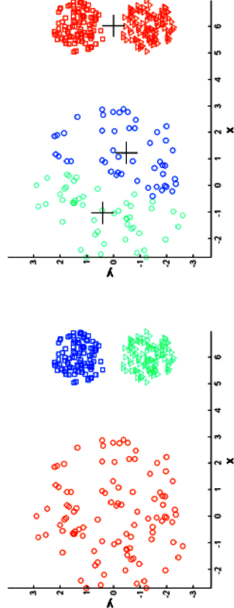
How to proceed?

- Choose a K (How?)
 - ▶ Run K-means algorithm multiple times
 - ▶ Choose clusters corresponding to the one that minimized sum of squared error (SSE)
- If $K = n$, no error.
- Good clustering has smaller K



²Hamery, Greg and Ekan, Charles, "Learning the k in k-means", pp 281–288, NIPS-2002

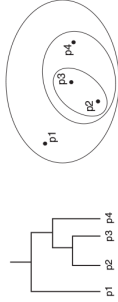
Limitations of K-means



- Has problem when data has
 - ▶ Different size clusters
 - ▶ Different densities
 - ▶ Non-globular shape
- Handling Empty Clusters
- When there are outliers
- Updating Centroids Incrementally

Other Approaches

- **Mini Batch K-Means** less computation and faster convergence
- **K-Medoids**: chooses data point as center and minimizes a sum of pairwise dissimilarities. Resistance to noise and/or outliers
- **Agglomerative Hierarchical Clustering**: repeatedly merging the two closest clusters until a single (Single Link)



- **DBSCAN**: density-based clustering algorithm that produces a partitional clustering, in which the number of clusters is automatically determined by the algorithm.
- **More variations**: Affinity propagation, Mean Shift, Spectral Clustering, Ward hierarchical, Optics, Gaussian Mixture, Birch

Evaluation of K-means

- **Choosing K**: 1) Domain Knowledge, 2) Preprocessing with another algorithm, 3) Iteration on K
- **Initialization of Centers**: 1) Random point in space, 2) Random point of data, 3) look for dense region, 4) Space uniformly in feature space, 5) K-Means++ (high probable if at far)
- **Cluster Quality**: 1) Diameter of cluster verses Inter-cluster distance, 2) Distance between members of a cluster and the cluster center, 3) Diameter of smallest sphere, 4) Ability to discover hidden patterns
- **Efficiently**: mini-batch K-Means

Important Note:

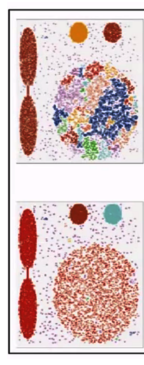
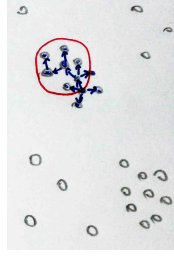
- K-Means and K-NN are different (K nearest neighbors)

K-NN is a supervised approach for classification

DBSCAN

DBSCAN³ (Density-Based Spatial Clustering of Applications with Noise) is a spatial clustering algorithm of KDD96

- Parameters (Eps/MinPts) and points (core/border/noise)
- Uses DFS

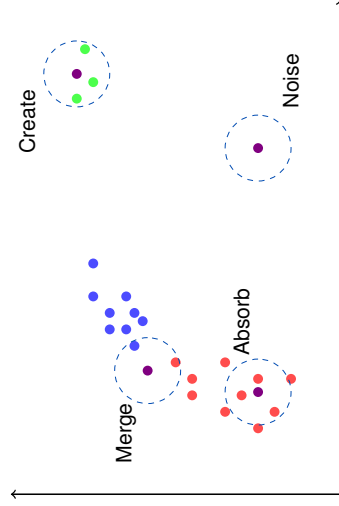


Figures from G. Karypis, E.-H. Han, and Y. Kumar, *COMPUTER*, 33(8), 1999

- Disadvantage: Sensitive to parameters
- Advantage: 1) clusters of arbitrary shape, 2) Can handle dynamic databases

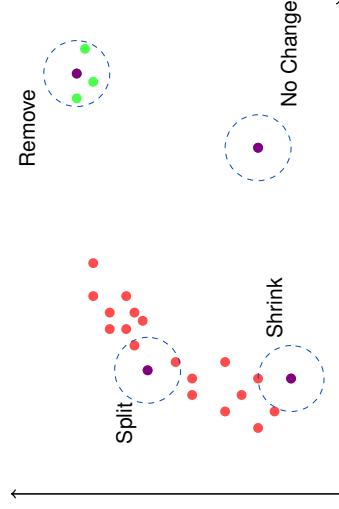
³A density-based algorithm for discovering clusters in large spatial databases with noise, M.Ester, HP Kriegel, J.Sander, X.Xu, KDD, 96(34), pp 226–231, 1996

Incremental DBSCAN (Addition)



Thank You!

Incremental DBSCAN (Deletion)



Thank you very much for your attention!

Queries ?