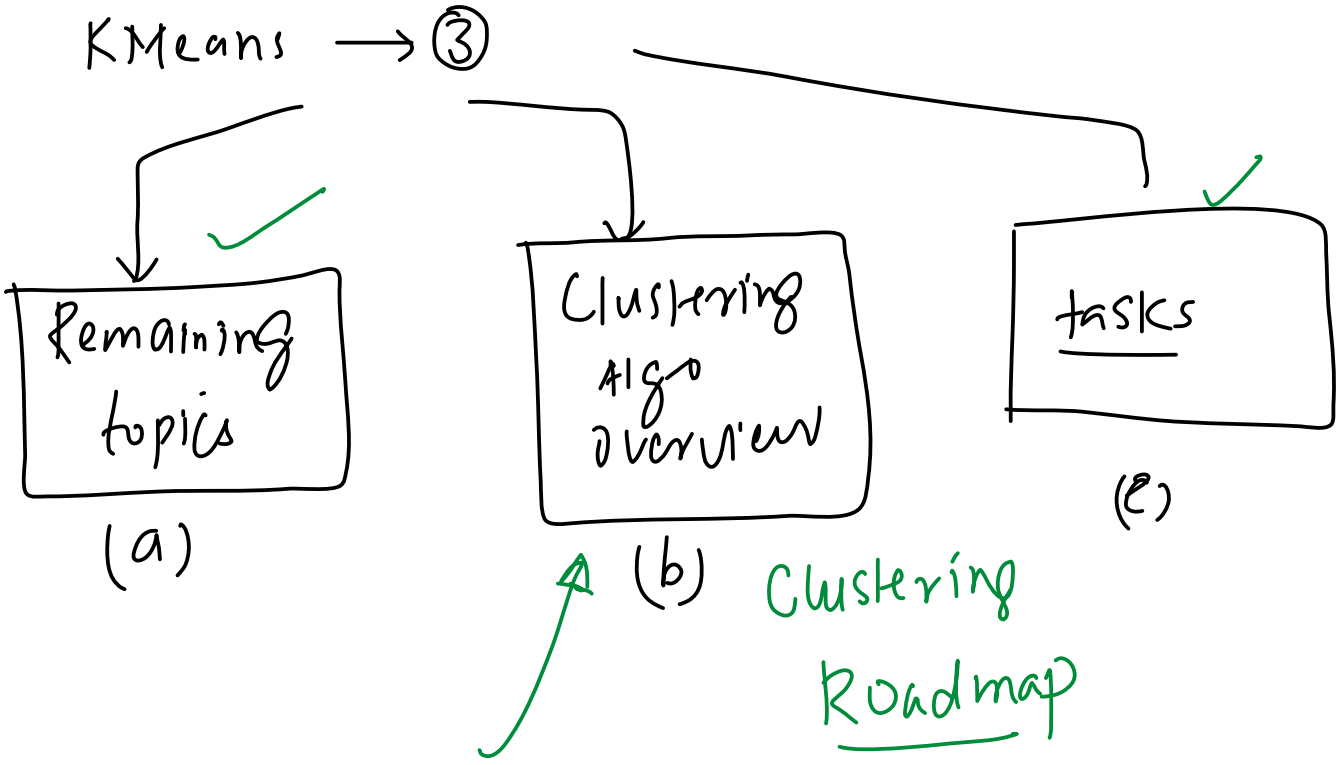
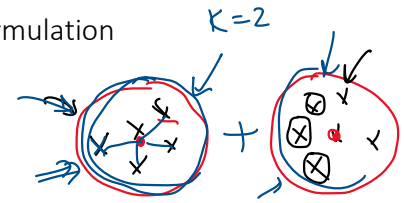
Recap

05 December 2023 17:15



Kmeans Mathematical Formulation

05 December 2023 17:16



Kmeans

[Lloyd's Algorithm] Jugaad

↓ inertia

↓ approximation

$$J = \underset{\mu_1 \mu_2 \dots \mu_K}{\operatorname{argmin}} \left[\sum_{i=1}^{K} \left[\sum_{x \in S_i} \left[\|x - \mu_i\|^2 \right] \right] \right]$$

such that

$[S_i \cap S_j = \{\}]$

Column inertia

$\mu_1 \mu_2 \dots \mu_K$ → centroid

K centroids

S_i → current cluster

K=2

$[S_i \cap S_j = \{\}]$

n

1000,

K

2 cluster

K

n points K cluster

$$J = \underset{\mu_1 \mu_2 \dots \mu_K}{\operatorname{argmin}} \left[\sum_{i=1}^{K} \sum_{x \in S_i} \|x - \mu_i\|^2 \right] \rightarrow \text{minimize}$$

such that $S_i \cap S_j = \{\}$

NP hard

non-convex → local minimal

approximation best soln

↓ very hard to solve

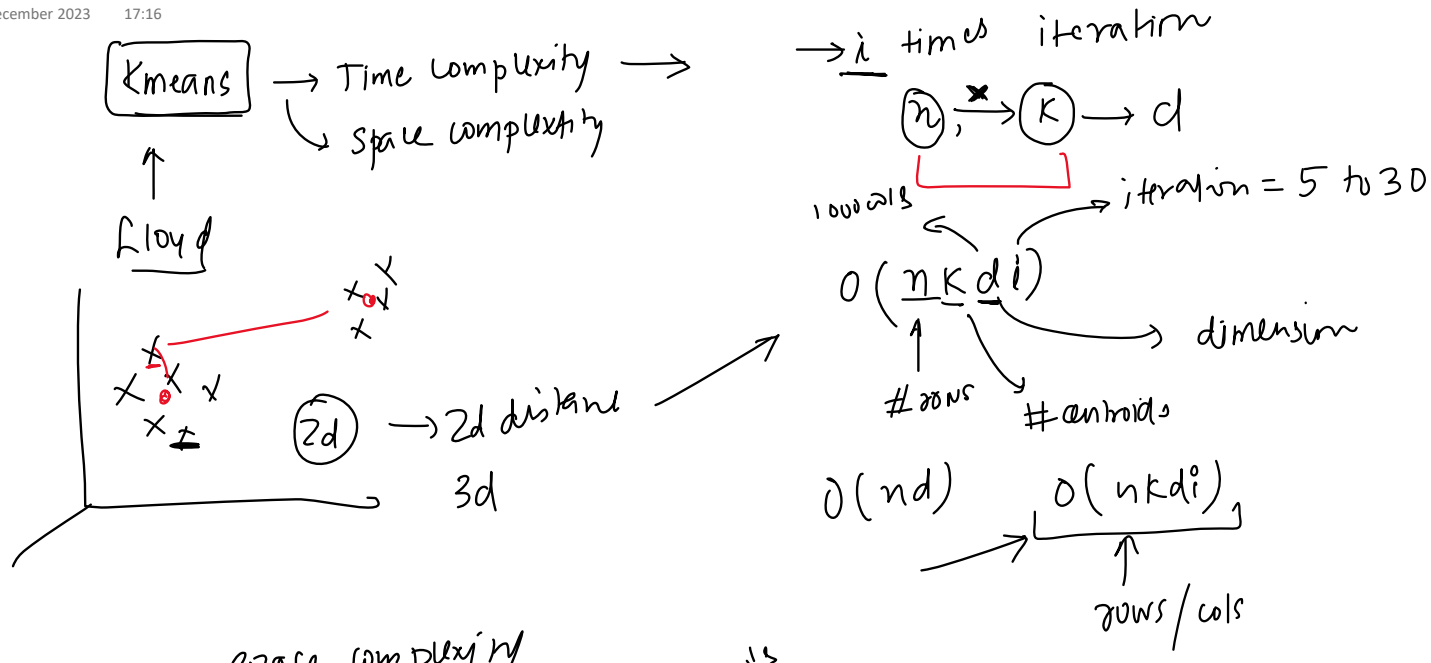
Jugaad

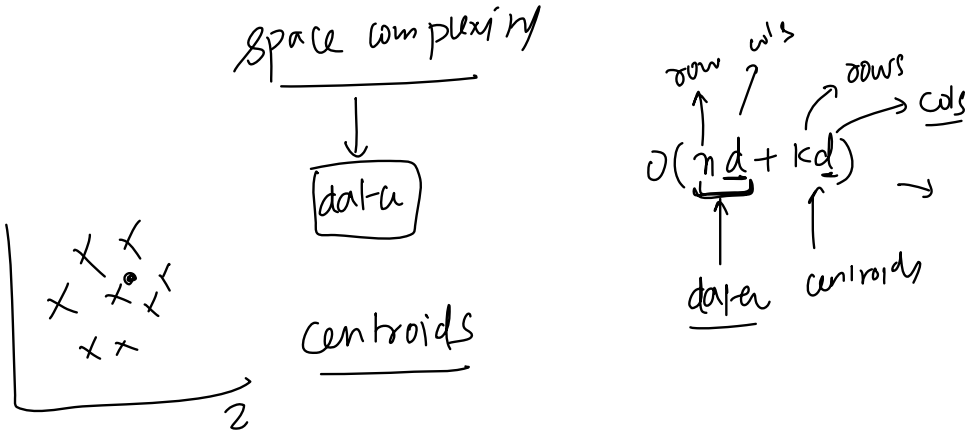
Lloyd

non-optimal local minima

Kmeans Time and Space Complexity

05 December 2023 17:16





Time → O(nKdi)

space → O(nd + Kd)

→ huge dataset

GBS

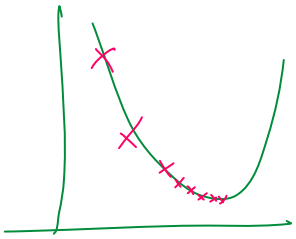
Mini Batch Kmeans

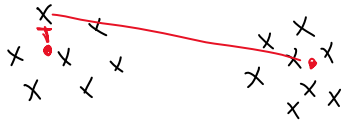
05 December 2023 17:15

Variation → Kmeans

↳ Mini Batch Kmeans

↳ Gradient Descent





→ n × K → distance

↓

batch

12 data 2 clust → 2 clusize

n = 4 × 2 → 8

batch size = 4

↳ mini batch

↳ speed

↳ Accura

Algorithm 1 Mini-batch k-Means.

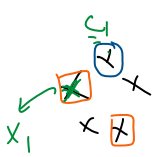
Given: k, mini-batch size b, iterations t, data set X
Initialize each $\mathbf{c} \in C$ with an $\mathbf{x}$ picked randomly from X
$\mathbf{v} \leftarrow 0$
for $i = 1$ to t **do**
  $M \leftarrow b$ examples picked randomly from X (mini batch)
  for $\mathbf{x} \in M$ **do**
    $\mathbf{d}[\mathbf{x}] \leftarrow f(C, \mathbf{x})$ // Cache the center nearest to $\mathbf{x}$
  end for
  for $\mathbf{x} \in M$ **do**
    $\mathbf{c} \leftarrow \mathbf{d}[\mathbf{x}]$ // Get cached center for this $\mathbf{x}$
    $\mathbf{v}[\mathbf{c}] \leftarrow \mathbf{v}[\mathbf{c}] + 1$ // Update per-center counts
    $\eta \leftarrow \frac{1}{\mathbf{v}[\mathbf{c}]}$ // Get per-center learning rate
    $\mathbf{c} \leftarrow (1 - \eta)\mathbf{c} + \eta\mathbf{x}$ // Take gradient step
  end for
end for

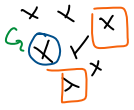
X → 200 points (n)

K → # clusters

b → 10

i → 10





K=3

K=2

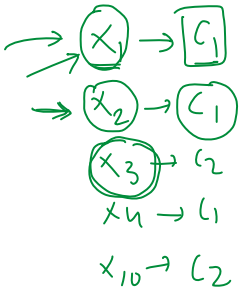
V = [0, 0]

loop → 1 − t / 10 times

[randomly 10 → 200 points] mini batch

loop → 10 times

→ [dict] (x, c)



10 data 2 clus

↓

20 distances

C_1

$V = [2, 0]$

$\eta = \frac{1}{2} = 0.5$

$C_1 = (1 - 0.5)\,C_1 + 0.5\,X_2$

C_2

$\rightarrow V = [2, 1]$

$\eta = 1$

→ loop → 10 times (X_1)

C_1 ①

$V = [1, 0]$

$\eta = \frac{1}{1} = 1$

$C_1 = (1 - \eta)\,C_1 + \eta X_1$

0

$C_1 = X_1$

$\eta = 1$

$C_1 = (1-0.5)\,C_1 + 0.5\,x_2$

$C_1 = x_1$

$C_2 = (1-1)\,C_2 + 1\,x_3$

$C_2 = x_3$

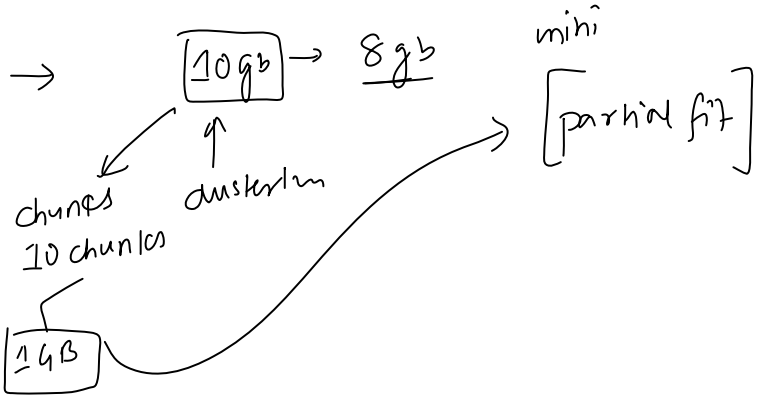
When should you use Minibatch

→ speeds / bigger dataset →

→ online ml

↓

Offline



downside

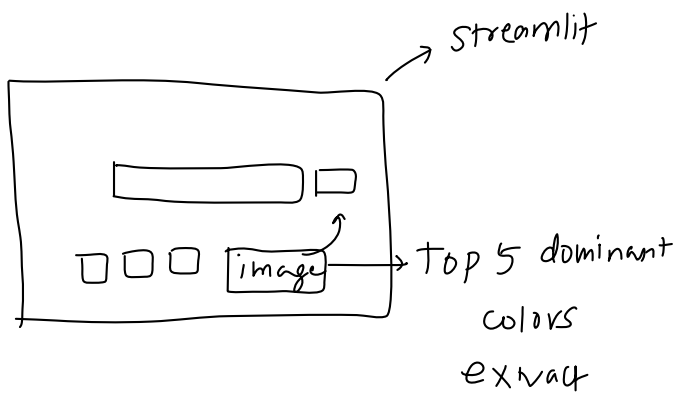
accuracy

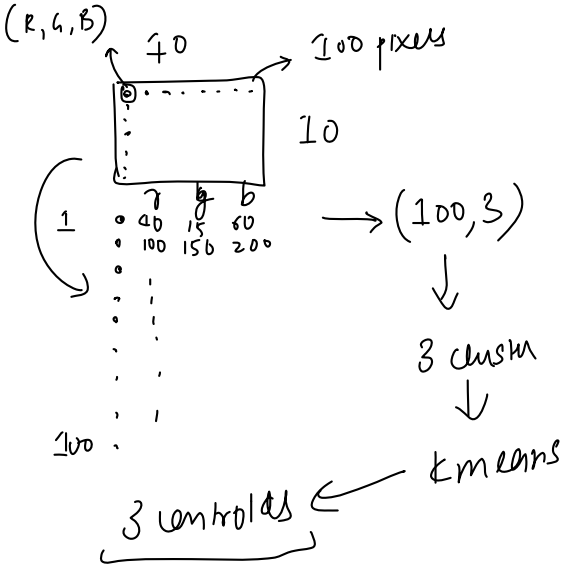
Task 1

05 December 2023 17:16

Task 2

05 December 2023 17:16

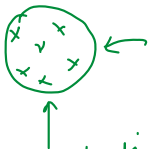




Types of Clustering

05 December 2023 10:57

given point n → k clusters

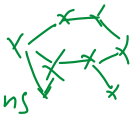


distribution

kmeans

Clustering

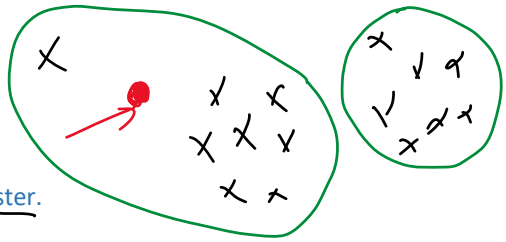
- Partitional clustering (1) ✓ → kmeans
- Hierarchical clustering (2) ✓
- Density based clustering (3) ✓ (vs centroid based) → DBSCAN
- Distribution based (model) (4) ✓ → Gaussian Mixture model
- Graph based clustering (5) → spectral clustering
- Soft clustering (vs hard clustering) (6) → Fuzzy Cmeans, GMM



Snahul ES

Partitional Clustering

05 December 2023 10:57

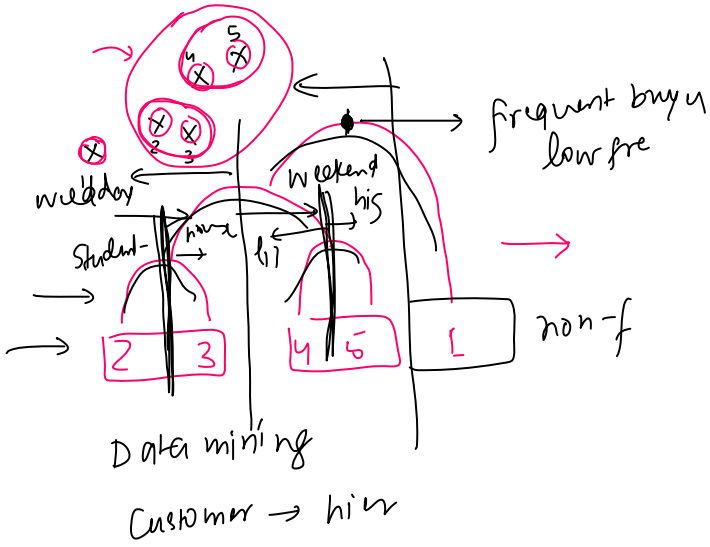


1. **Basic Concept**: Partitioning clustering algorithms divide a dataset into a set of non-overlapping subgroups or clusters, where each data point belongs to exactly one cluster.
2. **Examples**: The most famous partitioning clustering algorithm is k-means. It assigns data points to clusters in such a way that each point belongs to the cluster with the closest mean, which serves as a prototype of the cluster.
3. **Defining Number of Clusters**: A key requirement is pre-specifying the number of clusters (k). The selection of k significantly affects the outcome and quality of the clustering.
4. **Iterative Process**: These algorithms typically use an iterative refinement technique. For instance, in k-means, the process involves repeatedly assigning points to the nearest cluster centroid and then recalculating the centroids.
5. **Objective Function Optimization**: They aim to optimize an objective function, such as minimizing the total within-cluster variance or the sum of squared distances between data points and their respective cluster centroids.
6. **Suitability for Certain Data Shapes**: Partitioning methods are most effective when clusters are spherical or globular in shape. They assume homogeneity in cluster shapes and sizes.
7. **Sensitivity to Initial Conditions**: These algorithms can be sensitive to the initial starting conditions (like initial cluster centroids in k-means). Different initializations can lead to different clustering results.
8. **Handling of Outliers**: Partitioning algorithms can be influenced by outliers, as these can significantly skew the mean or centroid of a cluster.
9. **Scalability and Efficiency**: They are generally more scalable and efficient for larger datasets compared to hierarchical clustering, making them suitable for many practical applications.
10. **Use Cases and Limitations**: While widely used in various fields like market research, pattern recognition, and image processing, these algorithms have limitations in handling non-spherical clusters, varying cluster sizes, and noisy datasets. Advanced versions and variations of partitioning algorithms have been developed to address some of these limitations.

Hierarchical Clustering

05 December 2023 10:58

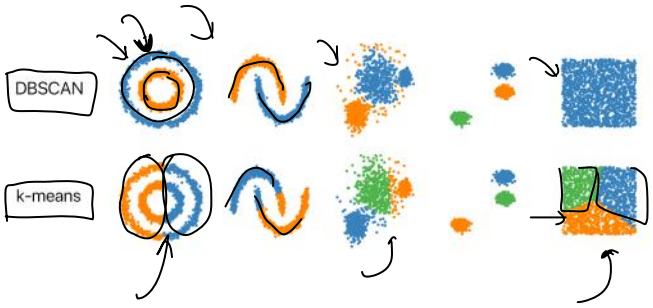
1. **Nature of Clustering**: Hierarchical clustering builds a hierarchy of clusters either by successively merging smaller clusters into larger ones (agglomerative approach) or by successively splitting larger clusters into smaller ones (divisive approach).
2. **No Need to Specify Number of Clusters**: Unlike partitioning algorithms like k-means, hierarchical clustering does not require pre-specifying the number of clusters. The number of clusters can be determined by analysing the dendrogram.
3. **Dendrogram Visualization**: It provides a tree-like diagram called a dendrogram, which is a visual representation of the clustering process showing the order of cluster combination and the distance at which clusters are merged.
4. **Distance Metrics and Linkage Criteria**: Hierarchical clustering uses various distance metrics (like Euclidean or Manhattan distance) and linkage criteria (like single linkage, complete linkage, average linkage, and Ward's method) to decide which clusters to merge or split.
5. **Flexibility in Identifying Cluster Shapes**: Hierarchical clustering can identify clusters with various shapes and sizes, unlike partitioning methods that generally assume spherical clusters.
6. **Computational Complexity**: It is generally more computationally intensive than partitioning methods, especially for large datasets, due to the need to compute and store distances between all pairs of points.
7. **Sensitivity to Noise and Outliers**: The method can be sensitive to noise and outliers, as these can influence the formation of clusters and the structure of the dendrogram.
8. **Applications**: It is widely used in fields like biology (for gene and protein sequencing), social science, and linguistics, and is particularly useful for exploratory data analysis where understanding the hierarchical relationship between objects is important.



Density Based Clustering

05 December 2023 10:58

- ✓ **Principle**: Density-based clustering groups data points based on the density of data points in a region. It defines clusters as areas of high density separated by areas of low density. The algorithm identifies clusters as regions where data points are densely packed together, with areas of low density or noise between them.
- ✓ **Examples**: DBSCAN is one of the most popular density-based clustering algorithms. It is known for its efficiency and ability to find clusters of arbitrary shapes.
- ✓ **No Need to Specify Number of Clusters**: Unlike partitioning methods, density-based clustering doesn't require pre-specifying the number of clusters.
- ✓ **Handling Noise and Outliers**: It is robust to outliers and noise, as these are typically not part of the dense regions that form clusters.
- ✓ **Ability to Find Arbitrary Shapes**: Density-based clustering can discover clusters of arbitrary shapes, unlike methods like k-means which are biased towards spherical clusters.
- ✓ **Parameter Sensitivity**: The performance of these algorithms is sensitive to the input parameters, like the radius of neighborhood (ε) and the minimum number of points required to form a dense region (MinPts in DBSCAN).
- ✓ **Scalability Issues**: Some density-based algorithms may struggle with very large datasets due to computational and memory constraints.
- ✓ **Applications**: Widely used in fields such as anomaly detection, geospatial data analysis (like identifying geographic regions of interest), and image processing, especially where the shape of the clusters is not known in advance or the data contains noise.



Distribution/Model based Clustering

05 December 2023 10:58

1. **Statistical Distribution Models**: The central concept of distribution-based clustering is that data points in a cluster follow a certain statistical distribution, most commonly Gaussian or normal distributions.
2. **Parameter Estimation**: These algorithms focus on estimating the parameters (like mean, variance) of the assumed distributions for each cluster. The fit of these parameters to the actual data determines the quality of the clustering.
3. **Expectation-Maximization (EM) Algorithm**: A key algorithm used in distribution-based clustering is EM, which alternates between assigning data points to the most likely distribution (expectation step) and updating the distribution parameters to maximize data fit (maximization step).
4. **Handling of Complex Cluster Shapes**: Unlike methods such as K-Means, distribution-based clustering can identify clusters of various shapes and sizes, making it more flexible in handling real-world data complexities.
5. **Computational Intensity**: The process of estimating distribution parameters and assigning data points can be computationally demanding, especially for large datasets and when the number of features (dimensions) is high.
6. **Handling of Outliers**: These methods can be more robust to outliers, as outliers are less likely to significantly affect the parameters of the overall distribution.
7. **Scalability Issues**: While effective for small to medium-sized datasets, scalability to very large datasets can be challenging due to the computational complexity of the algorithms and the need for more sophisticated optimization techniques.
8. **Soft Clustering**: In soft clustering, each data point is associated with a probability distribution across different clusters, indicating the degree of belonging to each cluster. This is in contrast to hard clustering, where each data point is assigned to exactly one cluster.