



DA unit -4 notes for One Shot video by brevilearning YT compressed 1

B.tech (Dr. A.P.J. Abdul Kalam Technical University)



Scan to open on Studocu

Data Analytics

UNIT - 4

[One - Shot]

Most important topics:

1. Mining frequent itemsets
2. Apriori algorithm
3. Flajolet - Martin (FM) algorithm.
4. K-means clustering techniques, Hierarchical clustering
5. CLIQUE and ProCLUS.

@brevilearning

Mining frequent itemsets:

- It involves finding reoccurring patterns, items or set of items in large datasets.
- This is crucial in fields like market basket analysis, where it helps identify items that frequently re-occur in transactions.

* Key concept:

- **Itemset:** A collection of one or more items.
e.g: { bread, butter, milk }
- **Support:** Frequency of occurrence of an itemset in data.
e.g: if { milk, butter } appears in 3 out of 10 transactions, its support is 30%.
- **Frequent itemset:** An itemset whose support is greater than or equal to a user-defined minimum threshold.

@brevilearning

Apriori Algorithm:

- It is a classic algorithm used in data mining for frequent itemsets and association rules.
- It uses bottom-up approach

where frequent subsets are extended one item at a time, and non-frequent subsets are pruned.

Steps of Apriori Algorithm:

Step 1: Generate candidate itemsets:

- Start with itemset of length 1.
- e.g: {bread}, {milk}, {butter}.

Step 2: Calculate support:

- Count the support for each candidate itemset.
- e.g: {bread} has 50%, {milk} has 20%.

Step 3: Prune non-frequent itemsets:

- Remove itemsets that do not meet the minimum support threshold.

- e.g: if minimum support is 30%, remove {milk} with 20% support.

@brevilearning

Step 4: Generate new candidates:

- Create new candidate itemsets by combining frequent itemsets.
- e.g: Combine {bread} and {butter} to form {bread, butter}.

Step 5: Repeat:

- Repeat the process for itemsets of increasing length until no more frequent itemsets can be generated.

Example: [Assume min. support threshold = 50%]

T1	{bread, milk}
T2	{bread, butter}
T3	{milk, butter}
T4	{bread, milk, butter}
T5	{bread, milk}

1. Generate candidate itemset :

{bread} , {milk} , {butter}

2. Calculate support for each:

{bread} : $4/5 = 80\%$

{milk} : $4/5 = 80\%$

{butter} : $3/5 = 60\%$

3. Frequent itemset :

all have support $\geq 50\%$

{bread} , {butter} , {milk}

4. Again generate candidate itemsets :

{bread , milk} , {bread , butter} , {milk , butter}

5. Calculate support for each :

{bread , milk} : $3/5 = 60\%$

{bread , butter} : $2/5 = 40\%$

{milk , butter} : $2/5 = 40\%$

6. Frequent itemset :

{bread , milk} (pruned other sets)

No further candidate will be generated since there are no frequent itemsets to extend.

Hence ,

frequent itemsets :

1. {bread} , {milk} , {butter}

2. {bread , milk}

@brownlearning

Flajolet - Martin (FM) Algorithm :

• It is a probabilistic algorithm used to estimate the no. of distinct elements in a data stream.

• It utilizes the use of hash functions and the properties of hash bit patterns in the no. of distinct elements to estimate the no. of distinct elements.

Algorithm :

Step 1: Hashing elements : For each element in the data stream, apply a hash func.

that maps the element to a binary number.

Step 2: Trailing zeros: For each hashed value, determine the no. of trailing zeros.

Step 3: Maximum trailing zeros: Track the max no. of trailing zeros observed. If the max no. of trailing zeros is R , then we estimate that there are approx 2^R distinct elements.

Step 4: Averaging (Multiple Hash functions): To improve accuracy, use multiple independent hash functions and take the average of their estimates.

@anujleaving

⑧ K-means clustering algorithm:

- It is a popular unsupervised machine learning algorithm used to partition a dataset into k -clusters, where each data point belongs to the cluster with the nearest

mean.

Algorithm:

1. Initialization:

- Choose the no. of clusters, k .
- Initialize k centroids randomly. These centroids can be randomly selected from the dataset.

2. Assignment step:

- Assign each data point to the nearest centroid. This forms k clusters.
- Distance is usually measured using the Euclidean distance formula.

3. Updation:

- Recalculate the centroid of each cluster.
- The new centroid is mean of all the points assigned to that cluster.

4. Reiteration :

- Repeat assignments and update steps until the centroid do not change significantly, or max no of iterations is reached.

@bnuilearning

⊗ Hierarchical clustering :

- It is a method of cluster analysis that seeks to build a hierarchy of clusters.
- It can be divided into two main types :
 1. Agglomerative (Bottom-up)
 2. Divisive (Top-down)

i) Agglomerative clustering :

- Starts with each data point as a single cluster.
- Iteratively merges the closest clusters until all points are in a single cluster or a desired

no. of clusters is reached.

★ Algorithm :

Step 1. Initialization :

- Start with n clusters (each data point is its own cluster)

Step 2. Compute proximity matrix :

- Calculate the distance (similarity) between every pair of clusters.

Step 3. Merge clusters :

- Find the pair of clusters with smallest distance.
- Merge these two clusters into a single cluster.

Step 4. Update proximity matrix :

- Recalculate the distances between the new cluster and all other clusters.

Step 5 Repeat :

- Repeat the merge and update steps until all points are merged into a single cluster or the desired no. of clusters is reached.

ii) Divisive clustering :

- Starts with all data points in one cluster.
- Iteratively splits the cluster into smaller clusters until each point is in its own cluster or a desired number of clusters is reached.

* Algorithm :

Step 1 Initialization :

- Starts with all points in one cluster.

SUBSCRIBE @brevilearning

Step 2. Split clusters :

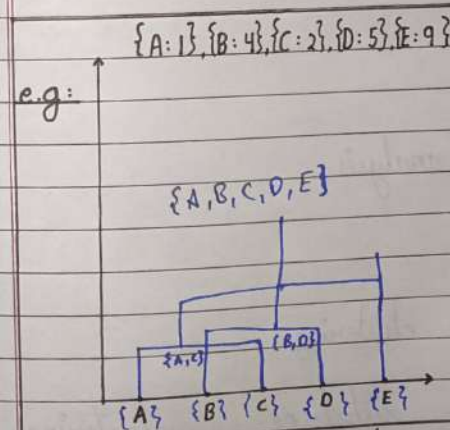
- Identify the best way to split the cluster into two sub-clusters. (mostly uses k-means clustering)

Step 3. Update clusters :

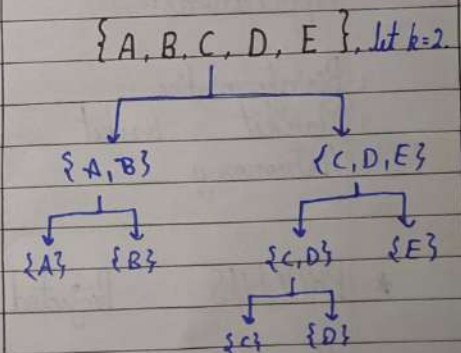
- Recalculate the distances b/w the new clusters and all other clusters (if applicable).

Step 4. Repeat :

- Repeat the split and update steps until each point is in its own cluster or the desired no. of clusters is reached.



Agglomerative Clustering



Divisive Clustering

CLIQUE and PROCLUS:

* CLIQUE: Clustering in Quest

- It is a powerful and efficient clustering algorithm designed for high-dimensional data.
- It combines grid-based and density based clustering approaches to discover clusters in subspaces of high-dimensional spaces.
- CLIQUE is notable for its ability to automatically determine subspaces that are relevant for clustering.

Applications:

1. Bioinformatics
2. Market basket analysis
3. Astronomy

* PROCLUS: Projected clustering

- It is a subspace clustering

algorithm specifically designed to handle high dimensional data.

- It identifies clusters in lower dimensional subspaces where clusters are not visible in full-dimensional space but appear in some lower dimensional projections.

Applications:

1. Bioinformatics
2. Customer segmentation
3. Text mining

Thanks for watching !!

@breuillearning