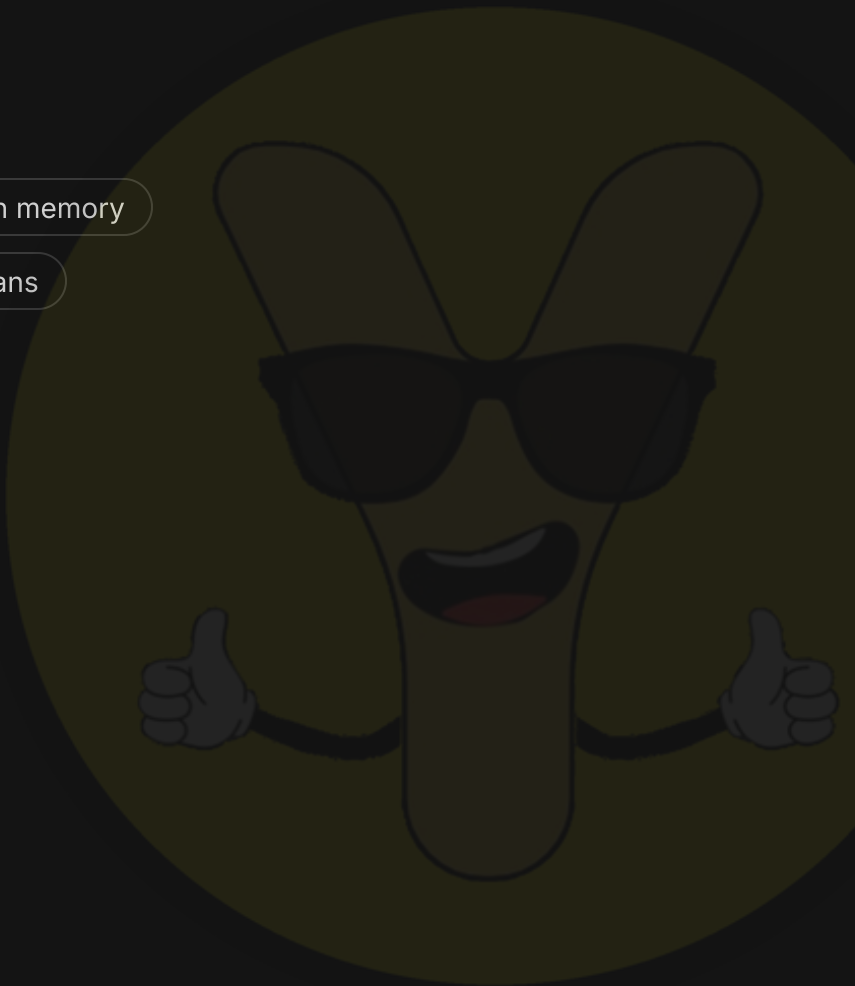
AKTU · 3RD YEAR · SEMESTER 6 · BADS601

Data Analytics

Unit 4 — Frequent Item Sets and Clustering

- Mining frequent item sets
- Market-basket modelling
- Apriori algorithm
- Handling large data sets in main memory
- Limited pass algorithm
- Counting frequent item sets in a stream
- Clustering techniques: hierarchical, K-means
- Clustering high dimensional data
- CLIQUE and ProCLUS
- Frequent pattern-based clustering methods
- Clustering in non-Euclidean space
- Clustering for streams and parallelism



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Mining Frequent Item Sets: Core Concepts

Frequent itemset mining is the foundational technique for discovering hidden patterns and co-occurrences in transactional datasets.

- Itemset: A collection of zero or more items denoted as $I = \{i_1, i_2, \ldots, i_m\}$.
- Transaction Database: A set of transactions $D = \{t_1, t_2, \ldots, t_n\}$, where each transaction $t_k \subseteq I$.
- Support Count: Absolute frequency of itemset X in D, given by $\sigma(X) = |\{t \in D \mid X \subseteq t\}|$.
- Relative Support: Fraction of transactions containing X, given by $\mathrm{Supp}(X) = \frac{\sigma(X)}{|D|}$.
- Frequent Itemset: An itemset whose support is greater than or equal to a user-defined threshold $s_{\min}$.
- Objective: Efficiently extract all itemsets satisfying $\mathrm{Supp}(X) \geq s_{\min}$ without exhaustive search.

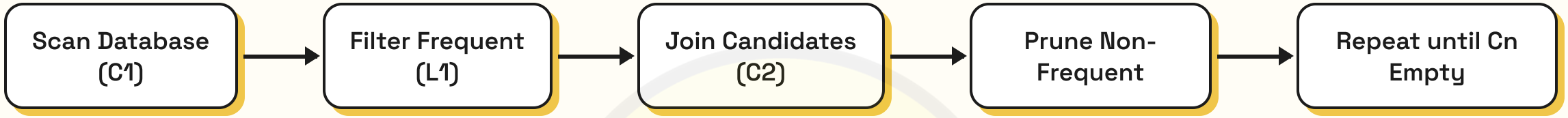
Market-Basket Modelling

MARKET-BASKET MODEL

A formal framework for analyzing customer buying habits by finding associations between items placed together in purchase baskets.

- Association Rule: An implication of the form $X \to Y$, where $X, Y \subseteq I$ and $X \cap Y = \emptyset$.
- Support: Measures rule popularity: $\text{Supp}(X \to Y) = \frac{\sigma(X \cup Y)}{|D|}$.
- Confidence: Measures rule reliability: $\text{Conf}(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$.
- Lift: Measures dependency strength: $\text{Lift}(X \to Y) = \frac{\text{Conf}(X \to Y)}{\text{Supp}(Y)}$.
- If Lift = 1, X and Y are independent; if Lift > 1, X and Y are positively correlated.

Apriori Algorithm: Candidate Generation & Pruning



Apriori relies on Anti-Monotonicity: If an itemset is infrequent, all of its supersets must also be infrequent.

Apriori Algorithm: Step-by-Step Calculation

Given dataset with T1:{A,B,C}, T2:{A,C}, T3:{A,D}, T4:{B,E,F}. Find frequent 2-itemsets for minimum support count s = 2.

1. Step 1: Count 1-itemsets (C1): {A}:3, {B}:2, {C}:2, {D}:1, {E}:1, {F}:1.
2. Step 2: Filter L1 (support >= 2): {A}:3, {B}:2, {C}:2.
3. Step 3: Generate candidate 2-itemsets (C2) by joining L1: {A,B}, {A,C}, {B,C}.
4. Step 4: Scan database to count support of C2: {A,B}:1, {A,C}:2, {B,C}:0.
5. Step 5: Filter L2 (support >= 2): {A,C}:2.

Frequent 2-itemset is {A,C} with Support Count = 2.

Handling Large Datasets in Main Memory

When transaction databases exceed RAM size, candidate 2-itemset counting (C2) creates a critical memory bottleneck.

- Memory Bottleneck: C_2 requires $O(n^2)$ memory spaces, where n is the number of distinct items.
- PCY (Park-Chen-Yu) Algorithm: Uses unallocated main memory in Pass 1 to build a hash table of candidate pairs.
- PCY Pass 1: Counts item frequencies and hashes candidate pairs $\{i, j\}$ to hash buckets.
- Bit Vector Reduction: Converts bucket counts to a bit vector using minimum support threshold $s_{\min}$.
- PCY Pass 2: Keeps pair $\{i, j\}$ in C_2 only if items i, j are frequent AND their hash bucket bit is 1.
- Multistage & Multihash: Extend PCY by applying multiple hash functions to further eliminate candidate pairs.

Limited Pass Algorithms Comparison

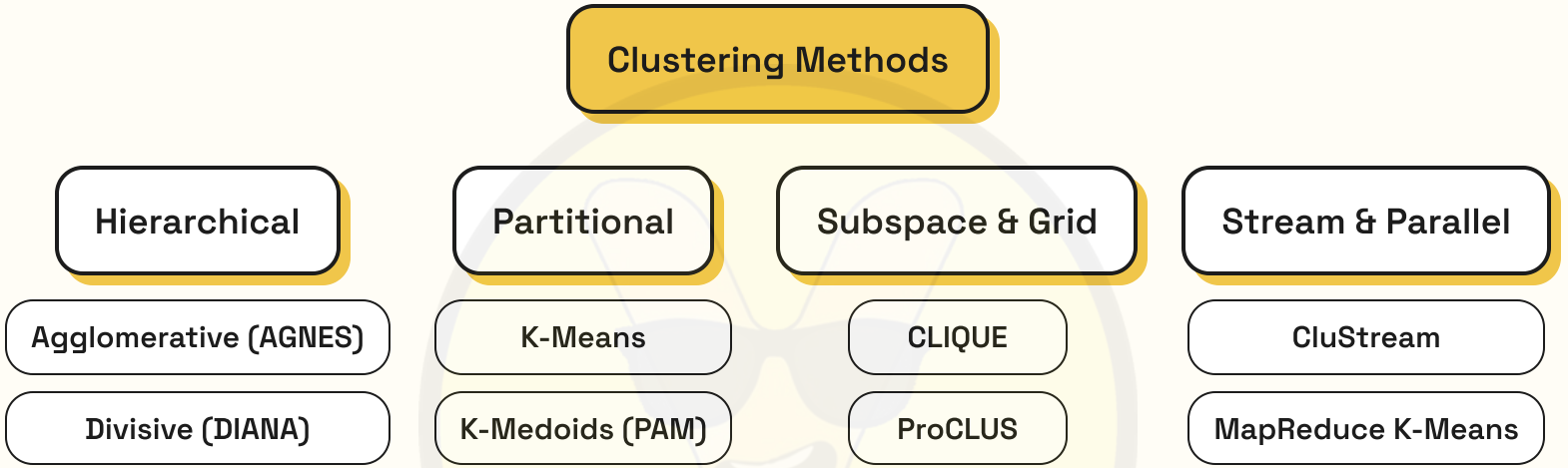
Algorithm	Passes	Core Mechanism	Trade-off / Property
Simple Randomized	1 Pass	Samples subset of data using scaled threshold $s' < s$	May produce false positives and false negatives
SON Algorithm	2 Passes	Divides data into chunks; itemset must be locally frequent in at least one chunk	Exact result; highly parallelizable using MapReduce
Toivonen Algorithm	2 Passes	Uses sample with lowered threshold and constructs a Negative Border	Exact result; succeeds with high probability in 2 passes

Counting Frequent Item Sets in Data Streams

Stream processing requires single-pass, real-time algorithms with bounded memory footprint.

- Stream Constraints: Continuous unbounded data, single-pass processing, limited memory, non-stationary distribution.
- Decaying Window Model: Assigns exponentially lower weights to older transactions using decay factor λ.
- Weight Function: Weight of item at time t is $w(t) = (1 - \lambda)^{t - t_0}$.
- Lossy Counting: Divides stream into dynamic buckets; drops itemsets whose estimated frequency falls below threshold.
- Sticky Sampling: Probabilistic algorithm that increases sampling probability for frequently seen stream items.

Taxonomy of Data Clustering Techniques



Clustering methods are broadly classified by structural hierarchy, partitioning strategy, dimensionality handling, and data scale.

Hierarchical Clustering Methods

Hierarchical algorithms construct a nested tree of clusters called a dendrogram without requiring predefined K.

- Agglomerative (AGNES): Bottom-up approach; starts with single-element clusters and repeatedly merges closest pairs.
- Divisive (DIANA): Top-down approach; starts with one single cluster containing all objects and recursively splits.
- Single Linkage: $d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$ (sensitive to chaining effect).
- Complete Linkage: $d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$ (favors compact spherical clusters).
- Average Linkage: $d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$.
- Dendrogram: Tree diagram representing the nested hierarchy and distance threshold of cluster merges.

K-Means Clustering Algorithm

K-MEANS CLUSTERING

A iterative partitional algorithm that divides N observations into K distinct, non-overlapping clusters based on distance to centroids.

- Objective Function: Minimize Sum of Squared Errors: $SSE = \sum_{k=1}^{K} \sum_{x \in C_k} \|x - \mu_k\|^2$.
- Step 1 (Initialization): Select K initial centroids $\mu_1, \mu_2, \ldots, \mu_K$ randomly or using K-Means++.
- Step 2 (Assignment): Assign each data point to its nearest centroid based on Euclidean distance.
- Step 3 (Update): Compute new centroid $\mu_k = \frac{1}{|C_k|} \sum_{x \in C_k} x$ for each cluster.
- Convergence: Repeat Assignment and Update until centroids stop changing or maximum iterations reached.

K-Means Algorithm: Worked Example

Cluster points {1, 2, 6, 7, 8} into K=2 clusters with initial centroids c1 = 2 and c2 = 7.

1. Iteration 1 (Assignment): Compare distance to c1(2) and c2(7): Point 1→c1, 2→c1, 6→c2, 7→c2, 8→c2.
2. Formed Clusters: C1 = {1, 2}, C2 = {6, 7, 8}.
3. Iteration 1 (Update): Compute new centroids: c1 = (1+2)/2 = 1.5, c2 = (6+7+8)/3 = 7.0.
4. Iteration 2 (Assignment): Re-assign points to c1(1.5) and c2(7.0): 1→c1, 2→c1, 6→c2, 7→c2, 8→c2.
5. Clusters unchanged: C1 = {1, 2}, C2 = {6, 7, 8}. Algorithm converged.

Final Clusters: C1 = {1, 2} (centroid 1.5), C2 = {6, 7, 8} (centroid 7.0).

Clustering High-Dimensional Data

High-dimensional data breaks traditional spatial clustering due to severe geometric distortions.

- Curse of Dimensionality: Distance to nearest neighbor approaches distance to furthest neighbor as dimensions increase.
- Sparsity: Data space becomes extremely sparse, rendering global density estimations meaningless.
- Irrelevant Features: Many dimensions contain noise that hides real structural clusters.
- Subspace Clustering: Locates clusters embedded in distinct subsets of total feature dimensions.
- Projected Clustering: Discovers specific lower-dimensional subspaces tailored to individual clusters.

CLIQUE Algorithm (Grid & Subspace)

CLIQUE (CLUSTERING IN QUEST)

A grid-based subspace clustering algorithm that automatically identifies dense regions in lower-dimensional subspaces using Apriori principles.

- Grid Partitioning: Divides each dimension into ξ non-overlapping interval units.
- Dense Unit: A grid cell is dense if the number of points inside exceeds density threshold τ.
- Monotonicity Property: If a k-dimensional cell is dense, all its $(k-1)$-dimensional projections are also dense.
- Step 1: Identify all dense 1-dimensional units.
- Step 2: Generate candidate k-dimensional dense units from $(k-1)$-dimensional dense units.
- Step 3: Join adjacent dense units to form maximal connected cluster regions.

CLIQUE vs ProCLUS Comparison

Parameter / Feature	CLIQUE Algorithm	ProCLUS Algorithm
Clustering Approach	Grid-based Subspace	Projected (Medoid-based) Subspace
Subspace Discovery	Bottom-up using Apriori grid candidate search	Top-down iterative assignment per cluster medoid
Cluster Overlap	Allows clusters to overlap across dimensions	Assigns each point to non-overlapping clusters
Parameters Required	Grid size ξ, Density threshold τ	Cluster count K, Average dimensions L
Dimensional Scalability	Scales exponentially with subspace dimension	Scales linearly with total number of dimensions

Frequent Pattern-Based Clustering

Pattern-based clustering groups high-dimensional text or transaction data using shared frequent patterns rather than spatial distance.

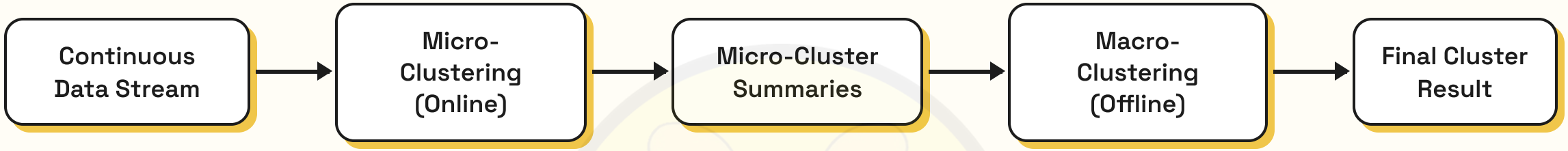
- Core Concept: Objects are clustered together if they share a common set of frequent itemsets or patterns.
- Application: Highly effective for document classification, gene expressions, and sparse market data.
- Cluster Overlapping: Data items can naturally belong to multiple clusters if they match multiple pattern sets.
- pCluster Algorithm: Extracts clusters of objects that exhibit similar pattern trends across feature subsets.
- MinApriori Clustering: Extracts frequent patterns to form cluster cores, avoiding spatial metric degradation.

Clustering in Non-Euclidean Space

Non-Euclidean spaces lack coordinate geometry, rendering vector means undefined.

- Distance Metrics: Uses Jaccard distance (sets), Cosine distance (text/vectors), or Edit distance (strings).
- Absence of Centroid: Physical arithmetic mean cannot be calculated without spatial coordinates.
- Clustroid: Selected representative data point that minimizes total distance to all other points in cluster.
- Clustroid Criteria: Point with minimum sum of squared distances, minimum max distance, or minimum average distance.
- BFR Algorithm: Extended K-means for massive Euclidean streams using summary statistics tuple $(N, SUM, SUMSQ)$.
- CURE Algorithm: Uses multiple representative points per cluster to discover arbitrary non-spherical shapes.

Stream Clustering Architecture (CluStream)



Stream clustering divides operations into fast online micro-clustering (summarization) and flexible offline macro-clustering.

Summary of Key Clustering Algorithms

Algorithm	Category	Time Complexity	Key Strength / Suitability
K-Means	Partitional	$O(n \cdot K \cdot i)$	Fast and efficient for spherical numerical data
Agglomerative	Hierarchical	$O(n^3)$	Produces complete dendrogram; no K required
CLIQUE	Subspace Grid	$O(n + 2^k)$	Finds dense clusters in arbitrary subspaces
ProCLUS	Projected Subspace	$O(n \cdot K \cdot L)$	Scales well for high-dimensional sparse data
CluStream	Data Stream	Online $O(1)$ per item	Handles real-time, evolving continuous streams
MapReduce K-Means	Parallel	$O(n/P)$ per node	Scales across distributed large-scale clusters

Exam Corner: Likely AKTU Questions

- **Q1** Define Support, Confidence, and Lift in Market-Basket Analysis. Derive formulas and explain their significance.
- **Q2** Explain the Apriori algorithm with a clear step-by-step example for a given transaction database.
- **Q3** Discuss the memory bottleneck in candidate 2-itemset generation. How does the PCY algorithm resolve it using hashing?
- **Q4** Differentiate between SON algorithm and Toivonen algorithm for limited pass frequent itemset mining.
- **Q5** Explain the K-Means clustering algorithm. Given a 1D dataset, trace 2 iterations showing centroid updates.
- **Q6** What is the Curse of Dimensionality in clustering? Compare CLIQUE and ProCLUS algorithms with neat feature comparisons.
- **Q7** Define Clustroid. Explain how clustering is performed in non-Euclidean space using non-metric distances.
- **Q8** Describe the online/offline framework of CluStream algorithm for counting and clustering continuous data streams.

Unit 4 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 5 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

