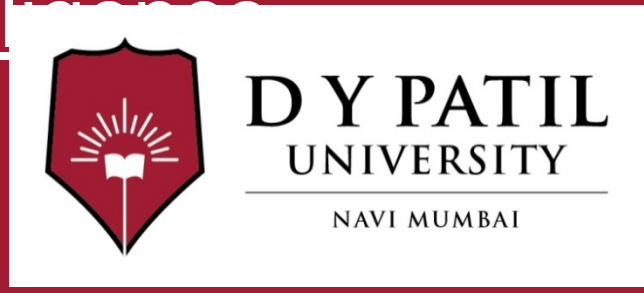
- Data Mining & Business Intelligence
- Module 4:Clustering



Index -

Lecture 20

Cluster Analysis

What is Cluster Analysis?

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups

- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters

- Unsupervised learning: no predefined classes

- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Clustering: Application Examples

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Climate: understanding earth climate, find patterns of atmospheric and ocean

Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
  - Distance-based (e.g., Euclidian) vs. connectivity-based (e.g., density)
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

Major Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Lecture 21

Partitioning Method

Partitioning Methods

- Partitioning method: Partitioning a database **D** of **n** objects into a set of **k** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)
- Global optimal: exhaustively enumerate all partitions
- Heuristic methods: k-means and k-medoids algorithms

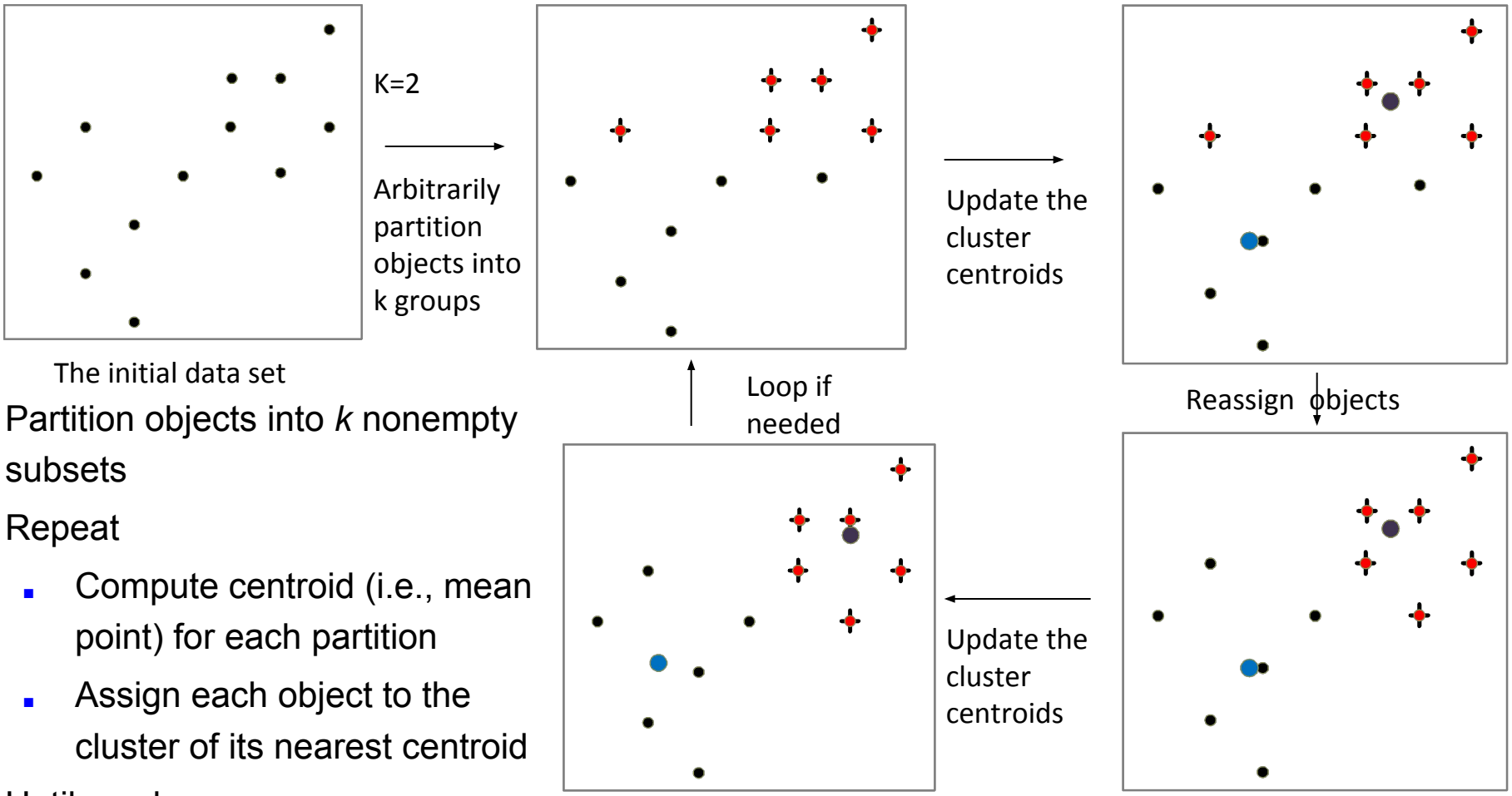
  Construct various partitions and then evaluate them by some criterion methods:

  1. k-means
  2. k-medoids
  3. CLARANS

The K-Means Clustering Method

- Given k, the k-means algorithm is implemented in four steps:
  1. Partition objects into k nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., mean point, of the cluster)
  3. Assign each object to the cluster with the nearest seed point
  4. Go back to Step 2, stop when the assignment does not change

An Example of K-Means Clustering



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster

Difference between K means and K mediods

- Both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups).
- K-means attempts to minimize the total squared error, while k-medoids minimizes the sum of dissimilarities between points labeled to be in a cluster and a point designated as the center of that cluster.
- In contrast to the k-means algorithm, k-medoids chooses data points as centers (medoids or exemplars).
- A mediod can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal. I.e., it is a most centrally located point in the cluster.

K mediods algorithm

- PAM, a k-medoids algorithm for partitioning based on medoid or central objects.
- Input: k: the number of clusters,
- D: a data set containing n objects.
- Output: A set of k clusters.
- Method: (1) arbitrarily choose k objects in D as the initial representative objects or seeds;
- (2) repeat
- (3) assign each remaining object to the cluster with the nearest representative object;
- (4) randomly select a nonrepresentative object, orandom;
- (5) compute the total cost, S, of swapping representative object, oj , with orandom;
- (6) if S < 0 then swap oj with orandom to form the new set of k representative objects;
- (7) until no change;

CLARA((Clustering LARge Applications)

- k-medoids partitioning algorithm like PAM works effectively for small data sets, but does not scale well for large data sets.
- To deal with larger data sets, a sampling-based method called CLARA (Clustering LARge Applications) can be used.
- Instead of taking the whole data set into consideration, CLARA uses a random sample of the data set.
- The PAM algorithm is then applied to compute the best medoids from the sample.
- Ideally, the sample should closely represent the original data set. In many cases, a large sample works well if it is created so that each object has equal probability of being selected into the sample.

CLARA((Clustering LARge Applications)

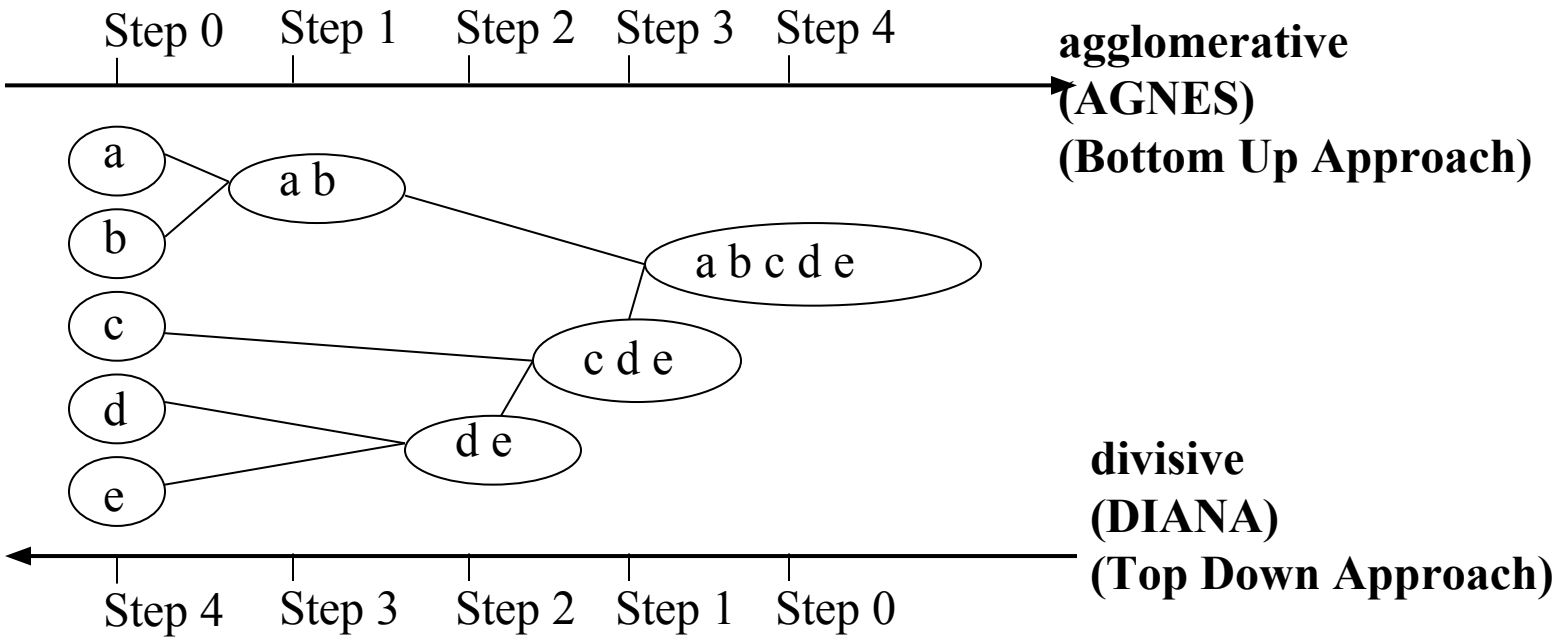
- The representative objects (medoids) chosen will likely be similar to those that would have been chosen from the whole data set.
- CLARA builds clusterings from multiple random samples and returns the best clustering as the output.
- The complexity of computing the medoids on a random sample is O(ks2 + k(n − k)), where s is the size of the sample, k is the number of clusters, and n is the total number of objects. CLARA can deal with larger data sets than PAM.

Lecture 22

Hierarchical Method

Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



Agglomerative Hierarchical Clustering

Given a set of n instances to be clustered, and an nxn distance (or similarity) matrix, the basic process hierarchical clustering is:

1 Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item.
Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.

2 .Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.

3. Compute distances (similarities) between the new clusters and each of the old clusters.

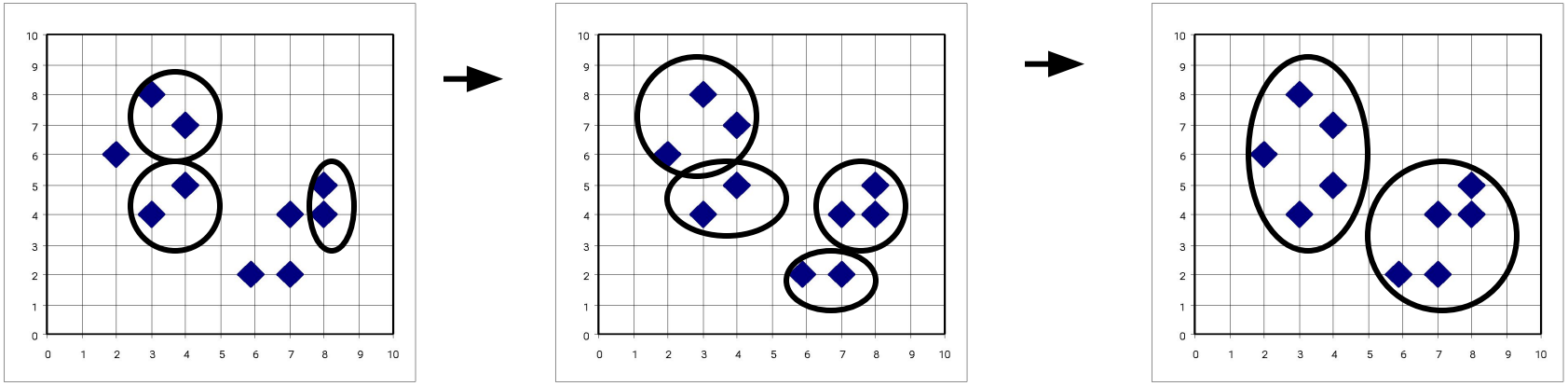
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size n.

More on Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - do not scale well: time complexity of at least $O(n^2)$, where n is the total number of instances
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster

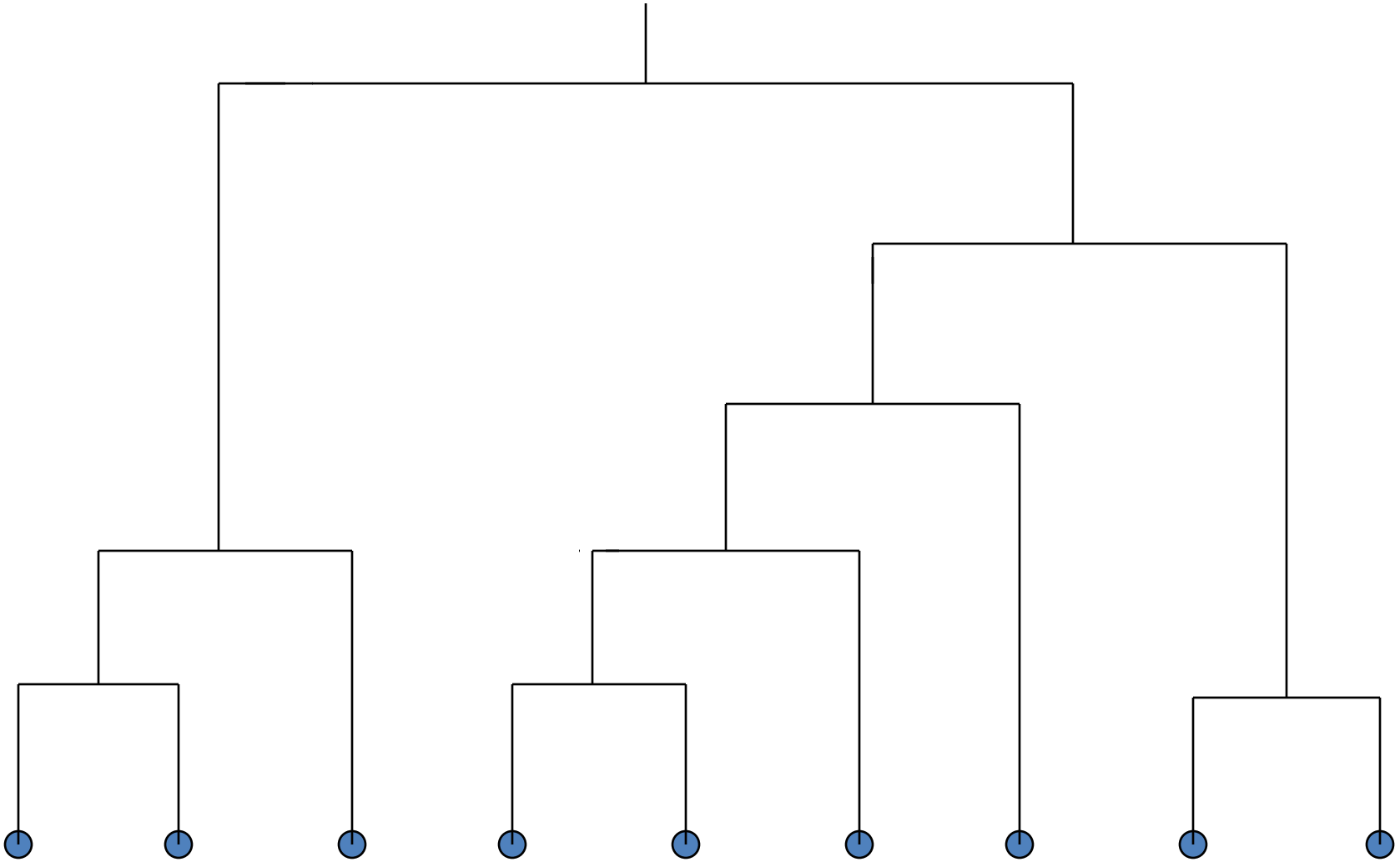


Dendogram

- Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

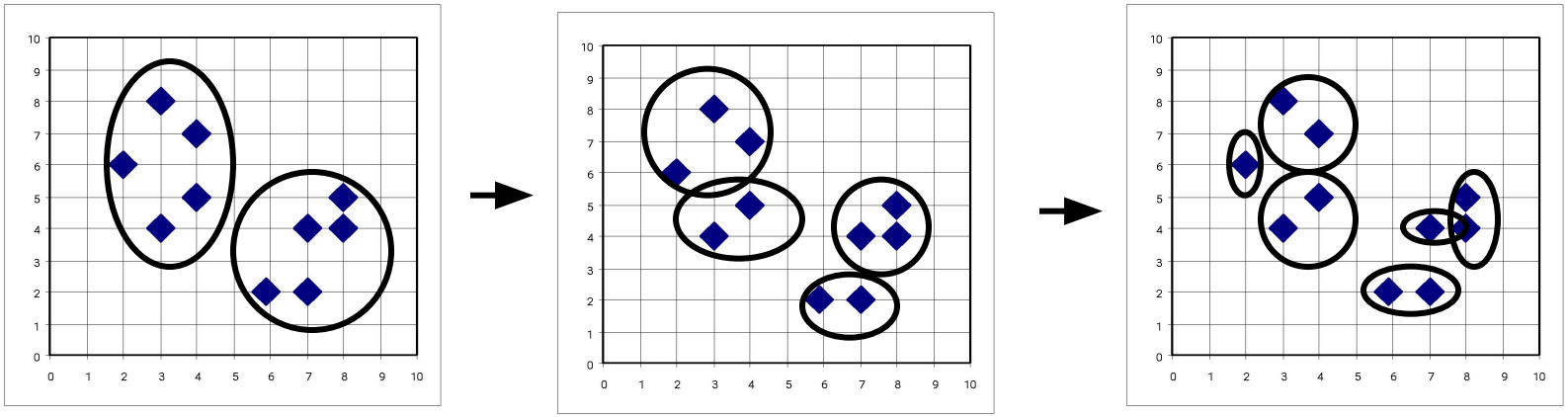
- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

A Dendrogram Shows How the Clusters are Merged Hierarchically



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



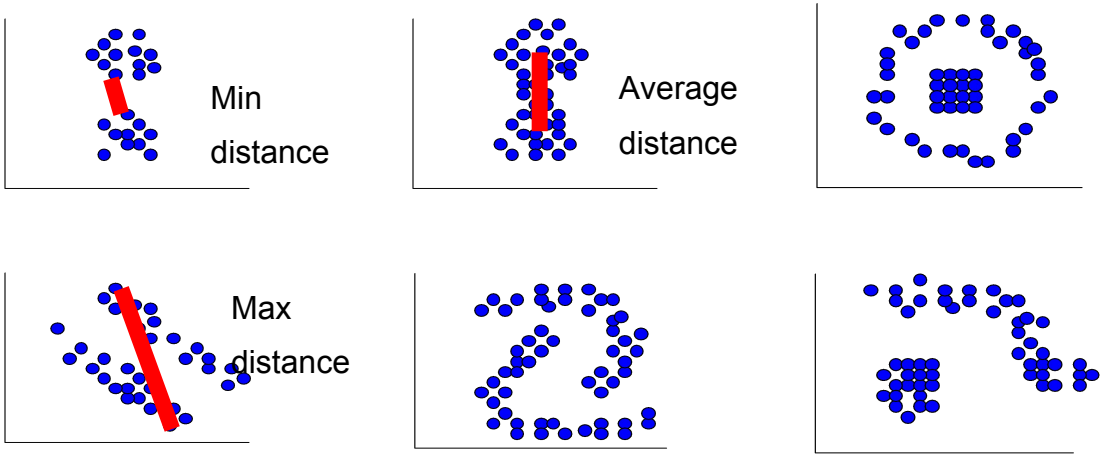
Computing Inter-Cluster Distances

- **single-link clustering** (also called the connectedness or minimum method) : we consider the distance between one cluster and another cluster to be equal to the **shortest distance** from any member of one cluster to any member of the other cluster. If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

- **complete-link clustering** (also called the diameter or maximum method): we consider the distance between one cluster and another cluster to be equal to the **longest distance** from any member of one cluster to any member of the other cluster.

- **average-link clustering** : we consider the distance between one cluster and another cluster to be equal to the **average distance** from any member of one cluster to any member of the other cluster.

Computing Inter-Cluster Distances



- ❑Single-Link Method / Nearest Neighbor
- ❑Complete-Link / Furthest Neighbor
- ❑Their Centroids.
- ❑Average of all cross-cluster pairs.

Determine the Number of Clusters

- Empirical method
  - # of clusters: k ≈√n/2 for a dataset of n points, e.g., n = 200, k = 10
- Other methods:
  - Elbow method
  - Cross validation method

Measuring Clustering Quality

- 3 kinds of measures: External, internal and relative
- External: supervised, employ criteria not inherent to the dataset
  - Compare a clustering against prior or expert-specified knowledge using certain clustering quality measure
- Internal: unsupervised, criteria derived from data itself
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are, e.g., Silhouette coefficient
- Relative: directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm

Lecture 23

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies

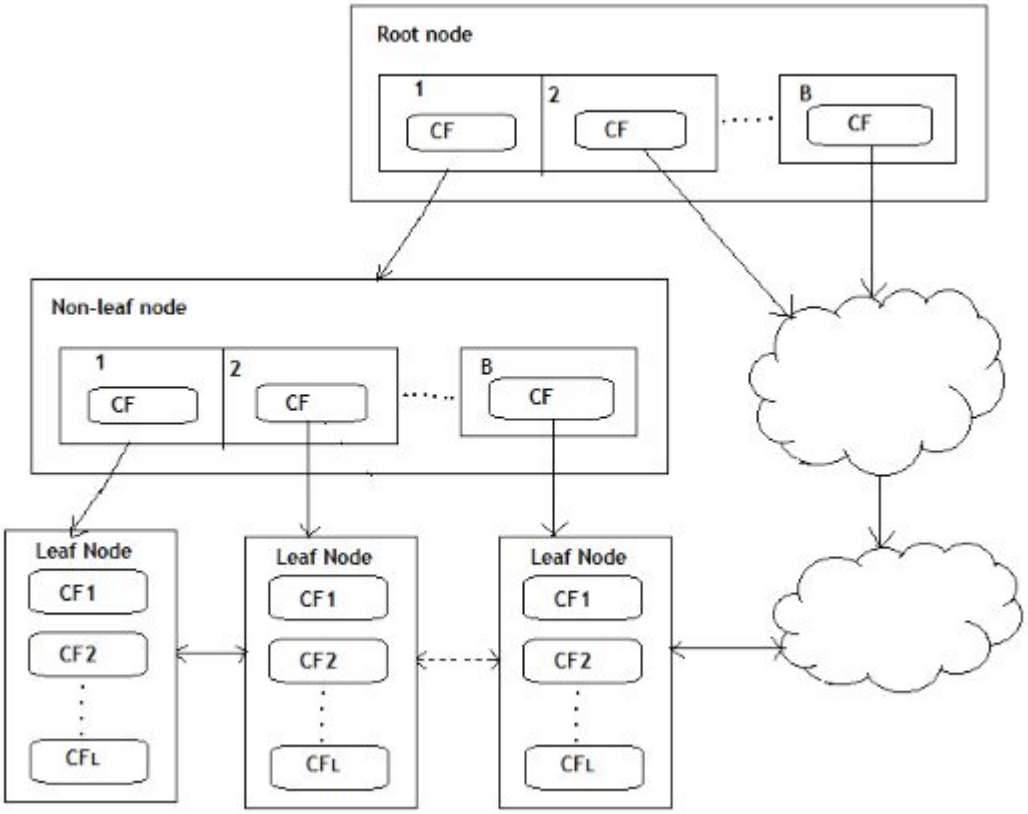
BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

- It is a scalable clustering method.
- Designed for very large data sets
- Only one scan of data is necessary
- It is based on the notation of CF (Clustering Feature) a CF Tree.
- CF tree is a height balanced tree that stores the clustering features for a hierarchical clustering.
- Cluster of data points is represented by a triple of numbers (N,LS,SS) Where
- N= Number of items in the sub cluster
- LS=Linear sum of the points
- SS=sum of the squared of the points

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

- **A CF Tree structure is given as below:**
- Each non-leaf node has at most B entries.
- Each leaf node has at most L CF entries which satisfy threshold T, a maximum diameter of radius
- P(page size in bytes) is the maximum size of a node
- Compact: each leaf node is a subcluster, not a data point

CF Tree Structure

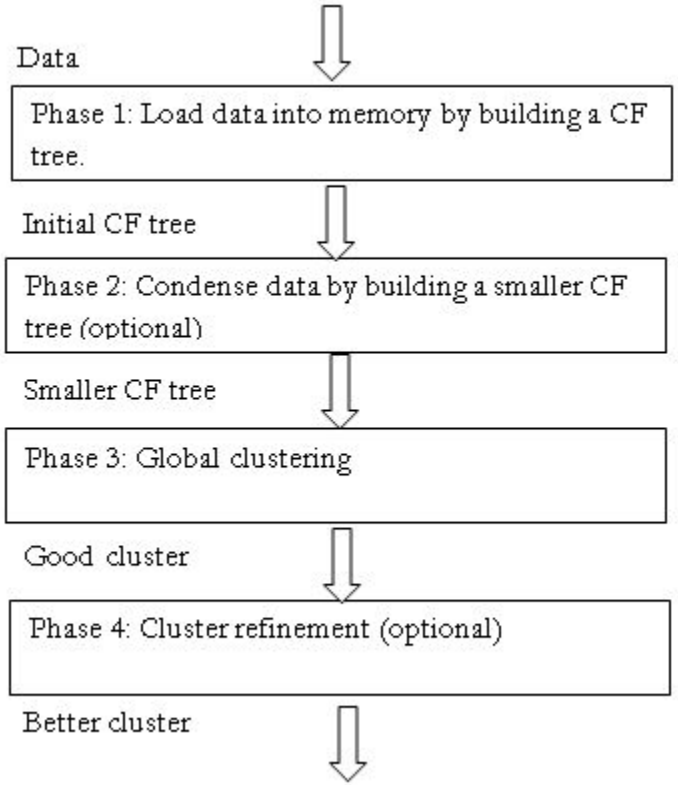


Basic Algorithm:

- Phase 1: Load data into memory
  - Scan DB and load data into memory by building a CF tree. If memory is exhausted rebuild the tree from the leaf node.

- Phase 2: Condense data(Compress Data)
  - Resize the data set by building a smaller CF tree
  - Remove more outliers
  - Condensing is optional
- Phase 3: Global clustering
  - Use existing clustering algorithm (e.g. KMEANS, HC) on CF entries
- Phase 4: Cluster refining
  - Refining is optional
  - Fixes the problem with CF trees where same valued data points may be assigned to different leaf entries.

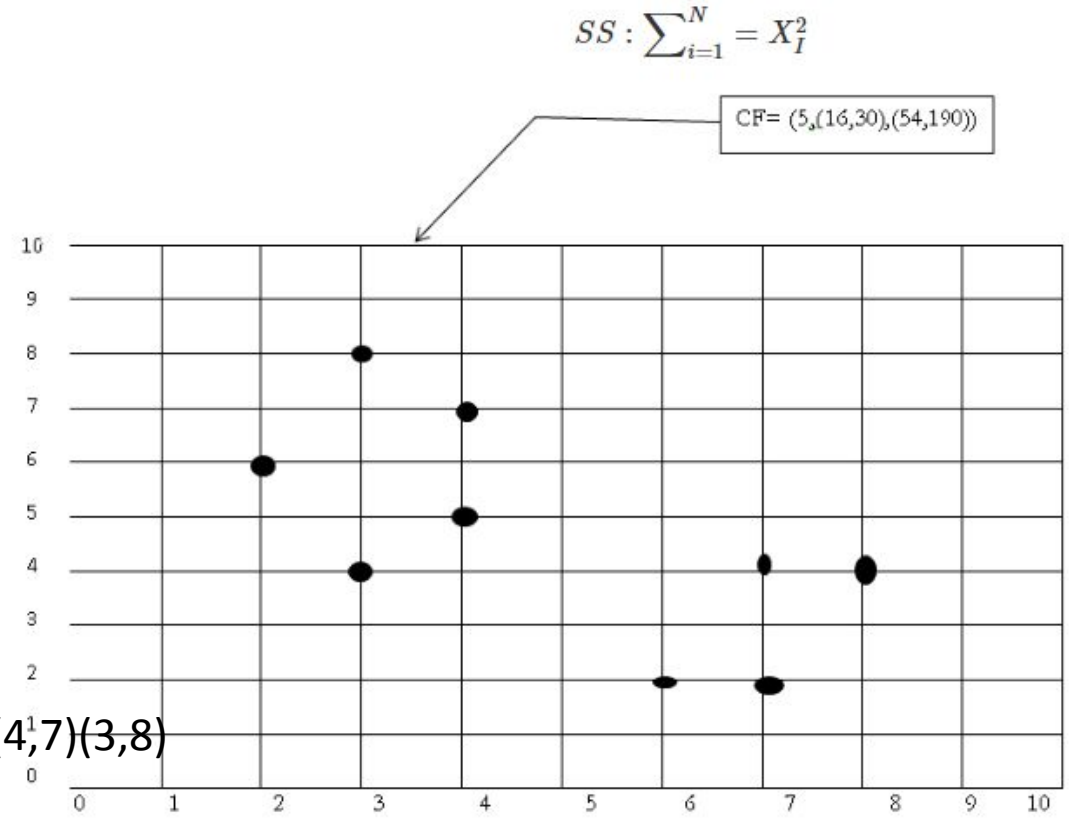
Basic Algorithm:



Example

- **Example:**
- **Clustering feature:**
- CF= (N, LS, SS)
- N: number of data points
- LS: ∑Ni=1=Xi
-

Example



(3,4) (2,6)(4,5)(4,7)(3,8)

N=5

NS= (16, 30) i.e. 3+2+4+4+3=16 and 4+6+5+7+8=30

SS=(54,190)=32+22+42+42+32=54 and 42+62+52+72+82=190

Advantages and Disadvantages

- Advantages: Finds a good clustering with a single scan and improves the quality with a few additional scans
- Disadvantages: Handles only numeric data
- Applications:
  - Pixel classification in images
  - Image compression
  - Works with very large data sets

Lecture 17

Density Based Clustering

DBSCAN

- DBSCAN is a density-based algorithm.
- DBSCAN requires two parameters: epsilon (Eps) and minimum points (MinPts).
- It starts with an arbitrary starting point that has not been visited .
- It then finds all the neighbour points within distance Eps of the starting point.
- If the number of neighbours is greater than or equal to MinPts, a cluster is formed.
- The starting point and its neighbours are added to this cluster and the starting point is marked as visited.
- The algorithm then repeats the evaluation process for all the neighbours recursively.

DBSCAN

- If the number of neighbours is less than MinPts, the point is marked as noise.
- If a cluster is fully expanded (all points within reach are visited) then the algorithm proceeds to iterate through the remaining unvisited points in the dataset.

Major features:

- Discover clusters of arbitrary shape
- Handle noise
- One scan
- Need density parameters

Basic concept:

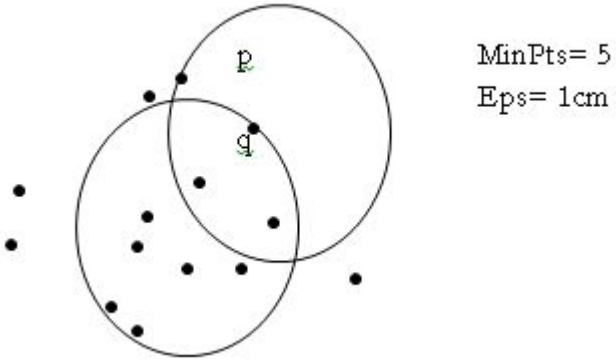
For any cluster we have:

- A central point (p) i.e. core point
- A distance from the core point(Eps)
- Minimum number of points within the specified distance (MinPts)

DBSCAN Algorithm

1. Create a graph whose nodes are the points to be clustered
2. For each core-point c create an edge from c to every point p in the □-neighborhood of c
3. Set N to the nodes of the graph;
4. If N does not contain any core points terminate
5. Pick a core point c in N
6. Let X be the set of nodes that can be reached from c by going forward;
   a. create a cluster containing X ∪ {c}
   b. N=N/(X ∪ {c})
7. Continue with step 4

DBSCAN Algorithm



MinPts: Minimum number of points in any cluster
ϵϵ : For each point in cluster there must be another point in it less than this distance away.
□-neighborhood: Points within □ distance of appoint
N ϵϵ(p) :{q belongs to D |dist(p,q) <=ϵϵ)
Core point: ϵϵ- neighborhood dense enough (MinPts)
Conditions: p belongs to N ϵϵ(q)
|Nϵϵ(q)| > = MinPts

Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points or based on an explicitly constructed density function
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - DENCLUE: Hinneburg & D. Keim (KDD'98/2006)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

Density-Based Clustering Methods

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius r (Eps)
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points