CS522 Advanced Database Systems
Clustering: Hierarchical Methods

Chengyu Sun
California State University, Los Angeles

# Hierarchical Clustering

- Agglomerative
  - Start with each object as a cluster
  - Recursively pick two clusters to merge
- Divisive
  - Start with all objects as a single cluster
  - Recursively pick one cluster to split

# Agglomerative Hierarchical Clustering

1. Compute a *distance matrix*
2. Merge the two *closest* clusters
3. Update the distance matrix
4. Repeat Step 2 until only one cluster remains

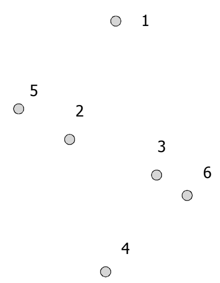
# Distance Between Clusters ...

- Min distance
  - Distance between two closest objects
  - Min < threshold: Single-link Clustering
- Max distance
  - Distance between two farthest objects
  - Max < threshold: Complete-link Clustering
- Average distance
  - Average of all pairs of objects from the two clusters

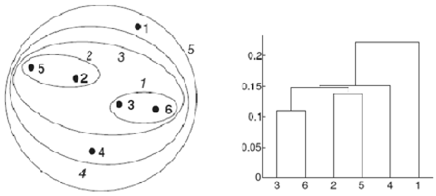
# ... Distance Between Clusters

- Centroid distance
- Increased SSE (Ward's Method)

# Min Distance Clustering Example ...



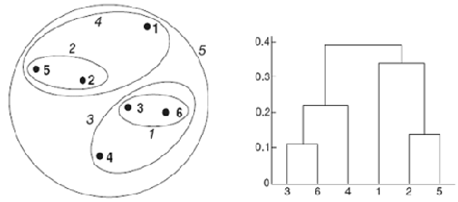
## ... Min Distance Clustering Example




(a) Single link clustering. (b) Single link dendrogram.



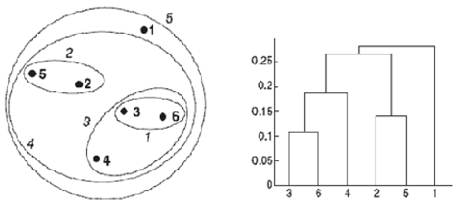
## Max Distance Clustering Example




(a) Complete link clustering. (b) Complete link dendrogram.



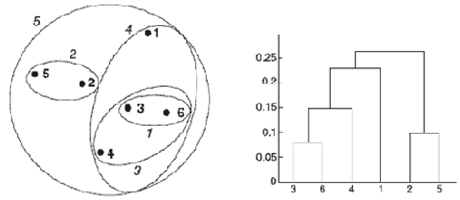
## Average Distance Clustering Example




(a) Group average clustering. (b) Group average dendrogram.



## Ward's Clustering Example




(a) Ward's clustering. (b) Ward's dendrogram.



## About Hierarchical Clustering

- Produces a hierarchy of clusters
- Lack of a global objective function
- Merging decisions are final
- *Expensive*
- Often used with other clustering algorithms

## BIRCH

- Balanced Iterative Reducing and Clustering using Hierarchies
  - Designed for clustering large amount of numerical data

## Clustering Feature (CF)

- A cluster can be represented by a clustering feature CF=<N, **LS**, SS>

N: number of objects

**LS** (Linear Sum): $\mathbf{LS} = \sum_{i=1}^{N} \mathbf{x}_i$

SS (Square Sum): $SS = \sum_{i=1}^{N} \mathbf{x}_i^2 = \sum_{i=1}^{N} \mathbf{x}_i \bullet \mathbf{x}_i$

## CF Example

- A cluster with two points (1,2) and (3,4)
  - N: 2
  - **LS**: (1+3,2+4) = (4,6)
  - SS: $1^2+2^2+3^2+4^2 = 30$

## Incremental Update of CF

- Cluster {(1,2), (3,4)}
  - Add a point (5,6)
  - Merge with cluster {(2,3),(4,5)}

## Using CF

- Centroid??
- Centroid distance??

## Cluster-to-Cluster Distances

- Cluster-to-cluster distances that can be calculated using CF
  - $D_0$: centroid Euclidean distance
  - $D_1$: centroid Manhattan distance
  - $D_2$: average inter-cluster distance
  - $D_3$: average intra-cluster distance
  - $D_4$: variance increase distance

## Cluster Diameter

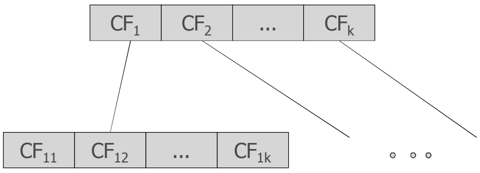
$$D = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(\mathbf{x}_i - \mathbf{x}_j)^2}{N(N-1)}} = \sqrt{\frac{2N \times SS - 2\mathbf{LS}^2}{N(N-1)}}$$

## About Clustering Feature

- Space efficiency
- Computation efficiency

## CF Tree

- Hierarchical clustering through tree construction (as oppose to agglomeration/division)




## CF Tree Input

- Dataset
- Threshold Condition
  - Diameter $D$ of a leaf node cluster < $d$

## CF Tree Insertion

- Insert an object into its closest cluster in a leaf node
  - The object is added to the cluster if the resulting cluster does not violate the threshold condition
  - Otherwise the object is added as a new cluster by itself
- When a node is full, split it and rebalance the tree (similar to B+ Tree Insertion)

## CF Tree Howto's

- Find closest cluster
  - *Object-to-cluster distance*
- Insert object into a cluster
  - *Update CF*
  - Check threshold condition
    - *Calculate diameter*
- Split node and rebalance tree
  - Merge clusters that are close to one anther
    - *Cluster-to-cluster distance; calculate CF of the merged cluster*

## About BIRCH

- Single scan of data
  - CF tree is kept in memory
  - Size of the CF tree can be adjusted using the threshold value
- Cluster the leaf node clusters
  - More natural clusters
  - Sparse clusters detected as outliers
- Require the notion of centroid

## Readings

- Textbook 10.3.1, 10.3.2, and 10.3.3