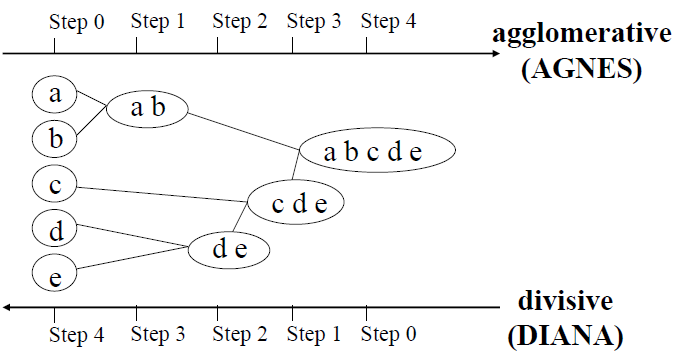
# Hierarchical methods

A method that creates a hierarchical decomposition of the given data objects. This method is either agglomerative (start with each object as its own cluster and then merge clusters) or divisive (start with all objects in one cluster and then divide). Example:



In the method, the distance matrix is used as clustering criteria and distance between clusters is defined by linage measures. In hierarchical clustering, we don’t have to specify number of clusters k as an input, but we need a termination condition (level in hierarchy)

AGNES (Agglomerative Nesting): Start with all objects in a cluster each. Then merge until all objects belong to the same cluster. Use single-link and dissimilarity matrix. Merge nodes with most similarity (least dissimilarity).

DIANA (Divisive analysis): start with a big cluster containing all objects. Then divide until each object is in its own cluster.

Major weakness of agglomerative clustering methods: Don’t scale well. Suffer from the fact that once a step (merge or split) is done, it can never be undone.

Better methods that integrate distance-based clustering into hierarchical:

* BIRCH – uses CF-tree and incrementally adjusts the quality of the sub-clusters
* ROCK – clustering categorical data by neighbor and link analysis
* CHAMELEON – hierarchical clustering using dynamic modeling

**BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)**

A method with two phases. It overcomes the problem in agglomerative methods that can’t undo what has been done. It also scales better.

BIRCH uses clustering feature (CF) to summarize a cluster and a CF-tree to represent a cluster hierarchy. The method incrementally constructs a CF-tree, the hierarchical data structure for multiphase clustering.

* Phase 1 – scan database to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
* Phase 2 – use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree

Weakness: only works with numeric data, sensitive to the order of the data record, and that we don’t always get natural clusters. Moreover, it has problem if the clusters are not spherical in shape since it uses the notion of diameter to control the boundary of a cluster

Clustering Feature:

From this we can compute the centroid of a cluster as

A CF is a summary of the statistics for a given sub-cluster and registers crucial measurements for computing the clusters. It also makes for less storage.

CF-tree:

A CF-tree is a height-balanced tree that stores the clustering features for a hierarchical clustering. The CF-tree is created with the CF’s. It has two parameters:

* B=branching factor, specifies the maximum number of children
* T=threshold, maximum diameter of sub-clusters stored at the leaf nodes

A nonleaf node in a tree has children and stores the sums of the CF’s of their children. A nonleaf node represents a cluster made of the sub-clusters represented by its children

A leaf node represents a cluster made of the sub-clusters represented by its entries.

Since each node in a CF-tree can only hold a limited number of entries due to its size, a CF-tree node does not always correspond to what we would think is a natural cluster.

New data point – check which centroid is the closest. Start at root node and go down to the leaf node. -> put object in closes one. Traverse the tree from root to find the leaf cluster which centroid is closest to the new point:

* If threshold condition OK

AND ENOUGH SPACE, add new object

* Otherwise, split (this means we might split natural clusters)
* Update the CF info

Split:

* Create new node
* Distribute data objects
* (Use farthest away objects as seeds for the clusters)
* Then connect the node to the parent
* If too many in parent, then split parent
* If no root, create root

**ROCK (Robust Clustering using linKs)**

Major ideas: Use links to measure similarity/proximity. Maximize the sum of the number of links between points within a cluster, minimize the number of links for points in different clusters.

Traditional measures for categorical data may not work well, therefore we use a link measure in ROCK.

Link measure in ROCK:

Points and are neighbors if and only if:   
where t and sim are between 0 and 1.

number of common neighbors between points and . The links between the points are the neighbors the points have in common. Decided by t.

Link measure for clusters: The number of cross links between clusters and

Goodness measure for merging clusters and :

Algorithm: sampling-based clustering. Stop algorithm when we’ve reached k clusters or goodness measure is zero

1. Draw random sample
2. Hierarchical clustering with links using goodness measure of merging
3. Label data in in disk: a point is assigned to the cluster for which it has the most neighbors after normalization