GRGPF Algorithm
(Ganti-Ramakrishnan-Gehrke-Powell-French)

Clustering in Non-Euclidean Spaces

Dealing With a Non-Euclidean Space

- **Problem: Clusters cannot be represented by centroid**
- **Why? Because the "average" of "points" may not be a point in the space.**
- **Solution: Use clustroid or a point in the cluster that minimizes the sum of the squares of distances to the points in the cluster.**

GRGPF Algorithm

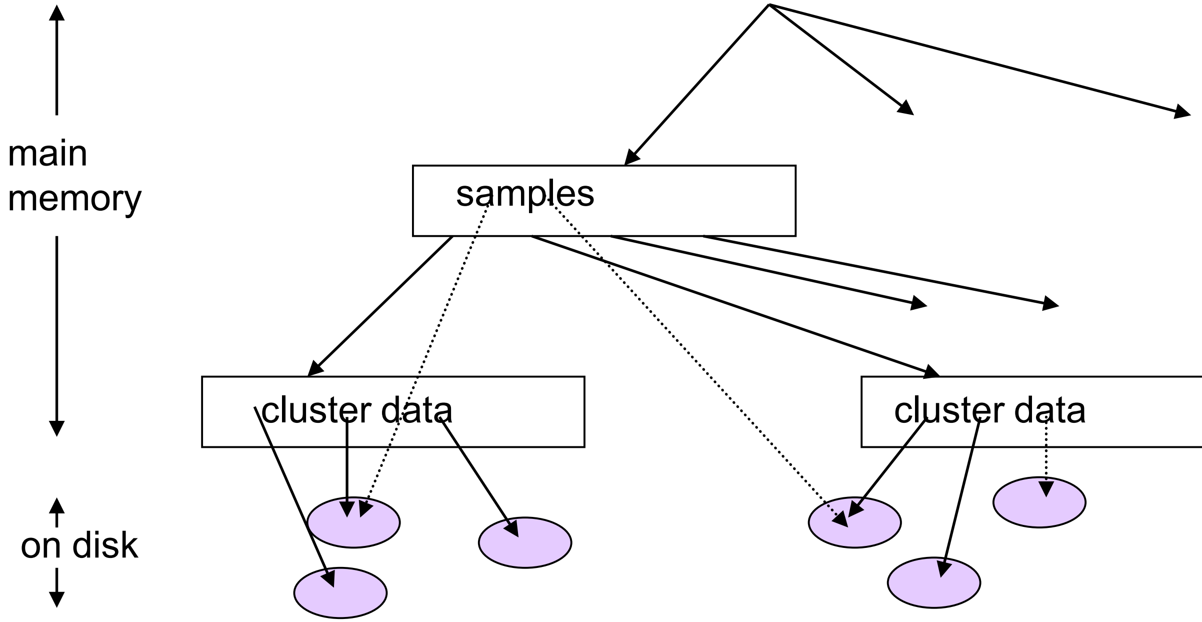
- **Ideas from both point assignment and hierarchical**
- **Represent clusters by sample points in memory**
- **Organize clusters hierarchically in a tree**
  - A new point is assigned to appropriate cluster by passing it down the tree
- **Leaves of tree hold summaries of some clusters**
- **Internal nodes hold information of clusters reachable through the nodes**
- **Group clusters by their distance from one another**
  - Clusters at a leaf are close
  - Clusters reachable from an interior node are relatively close

Representing Clusters in GRGPF

- **Information kept about a cluster**
  1. *N*, clustroid, SUMSQ (sum of the squares of the distances from clustroid to all points in the cluster)
  2. The *p* points closest to the clustroid, and their values of SUMSQ.
  3. The *p* points of the cluster that are furthest away from the clustroid, and their values of SUMSQ.

Interior Nodes of Tree in GRGPF

- Interior nodes have samples of the clustroids of clusters found at their descendant leaves.
- Try to keep clusters on one leaf block close, descendants of a level-1 node close, etc.
- Interior part of tree kept in main memory.



GRGPF Algorithm

Initialization

- Take a main-memory sample of points and cluster them hierarchically.
- Build the initial tree, with level-1 interior nodes representing clusters of clusters, etc.
- All other points are inserted into this tree.

Inserting Points

- Start at the root.
- At each interior node, visit the child node that have sample clustroid nearest the inserted point.
- At the leaf, insert the point into the cluster with the nearest clustroid.

GRGPF Algorithm

Updating Cluster Data

- **Suppose we add point *X* to a cluster.**
- **Increase count *N* by 1.**
- **For each of the 2*p* + 1 points *Y* whose SUMSQ is stored, add $d(X,Y)^2$.**
- **Estimate SUMSQ for *X*.**
  - If C is the clustroid, then SUMSQ(*X*) is

    $$\text{SUMSQ}(C) + N * d(X,C)^2$$

  - Assume that vector from *X* to *C* is perpendicular to vectors from *C* to all the other nodes of the cluster.
  - This value may allow *X* to replace one of the closest or furthest nodes.

GRGPF Algorithm

Possible Modification to Cluster Data

- **There may be a new clustroid --- one of the *p* closest points --- because of the addition of *X*.**
- **Eventually, the clustroid may migrate out of the *p* closest points, and the entire representation of the cluster needs to be recomputed.**

GRGPF Algorithm

Splitting and Merging Clusters

- **Maintain a threshold for the *radius* of a cluster**
  $= \sqrt{(\text{SUMSQ}/N)}$
- **Split a cluster whose radius is too large.**
- **What happens when we split so much that the tree no longer fits in main memory?**
  - Raise threshold on radius and merge clusters that are sufficiently close

GRGPF Algorithm

Merging Clusters

- **Suppose we have two nearby clusters with clustroids *D* and *E***
- **Compute SUMSQ(*X*) [from the cluster of *D*] for the combined cluster by summing:**
  1. SUMSQ(*X*) from its own cluster.
  2. SUMSQ(*E*) + $N\,[d(X,D)^2 + d(D,E)^2]$.
- **Point with the least SUMSQ is the clustroid for the combined cluster**
- **If the SUMSQ is too large, do not merge clusters.**
- **Hope to have enough mergers to fit tree in main memory.**

Summary

- Given a set of points, with a notion of distance between points, group the points into some number of *clusters*
- Centroid in Euclidean space and clustroid in non-Euclidean space
- Agglomerative hierarchical clustering
- *k*-means, BFR (*k*-means extended for large data sets),

  CURE (*k*-means extended for arbitrary clusters)
- GRGPF (clustering in non-Euclidean space)