Clustering

SNU. KDD LAB
Kyuseok Shim

CURE

- [Guha, Rastogi, Shim 98]
- Propose a new hierarchical clustering algorithm
  - Use a small number of representatives
  - Note:
    - Centroid-based: use 1 point to represent a cluster => Too little information..Hyper-spherical clusters
    - MST-based: use every point to represent a cluster =>Too much information..Easily mislead
- Use random sampling
- Use Partitioning
- Provide correct labeling

CURE

A Representative set of points:

- Small in number : c
- Distributed over the cluster
- Each point in cluster is close to one representative
- Distance between clusters:

smallest distance between representatives

CURE

Finding Scattered Representatives

- We want to
  - Distribute around the center of the cluster
  - Spread well out over the cluster
  - Capture the physical shape and geometry of the cluster
- Use farthest point heuristic to scatter the points over the cluster
- Shrink uniformly around the mean of the cluster

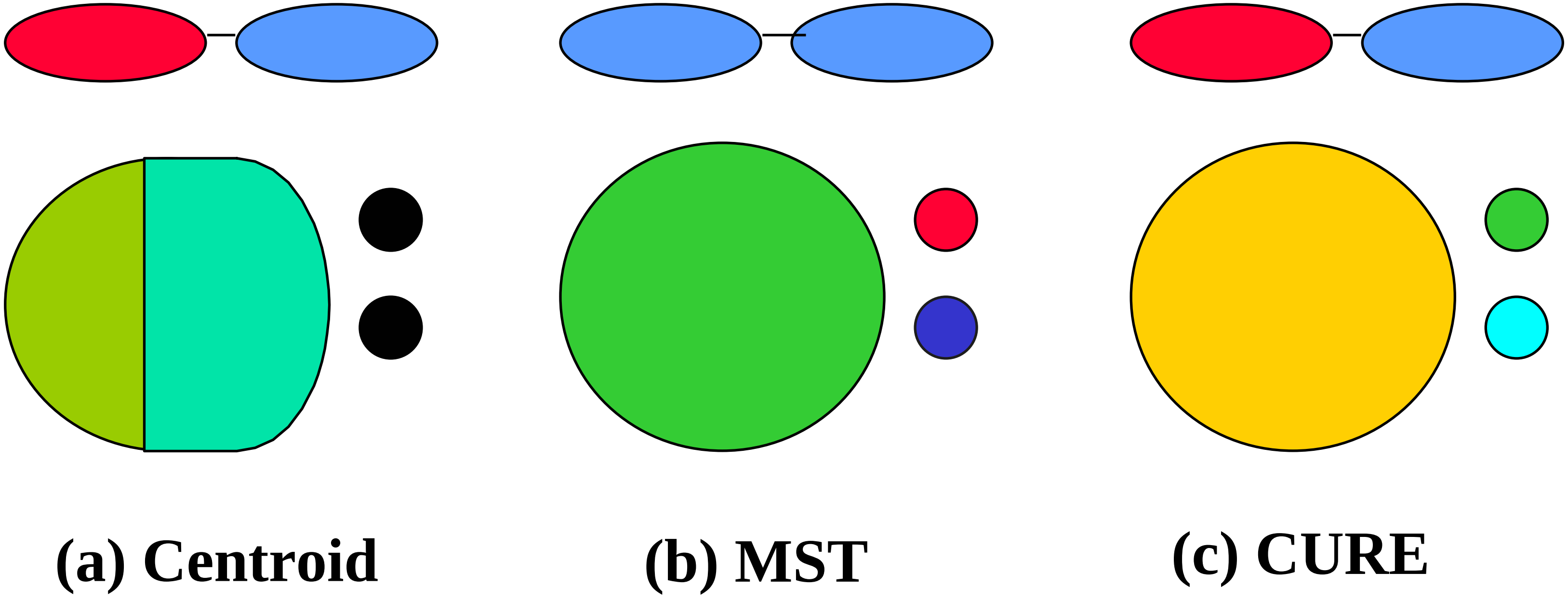
CURE

- Random sampling
  - If each cluster has a certain number of points, with high probability we will sample in proportion from the cluster
  - εn points in cluster translates into εs points in sample of size s

  Sample size is independent of n to represent all sufficiently large clusters
- Labeling data on disk
  - Choose some constant number of representatives from each cluster

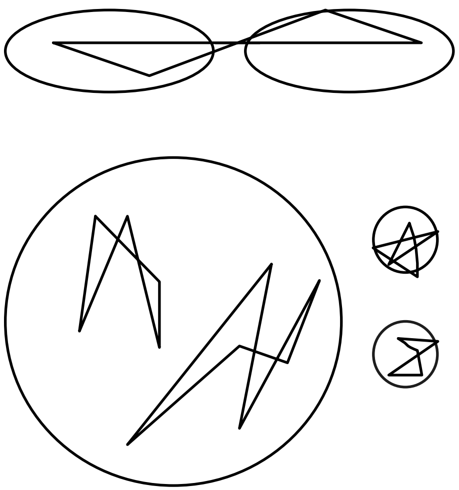
CURE

Comparisons

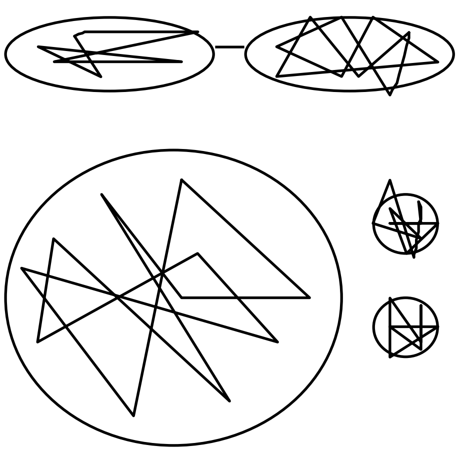


CURE

Number of Representatives



(a) c = 5

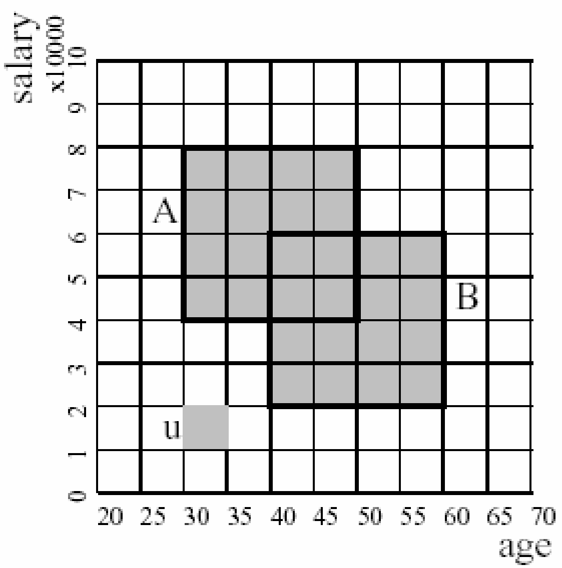


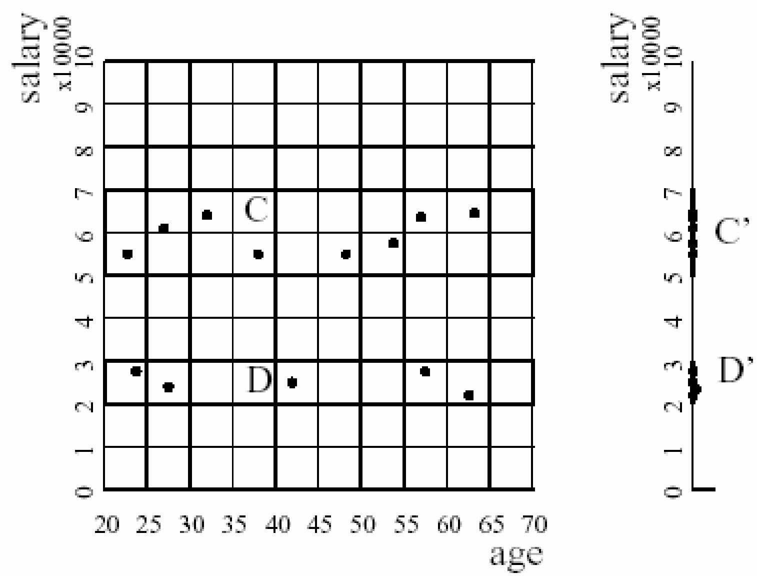
(b) c = 10

CLIQUE

- [Agrawal, Gehrke, Gunopulos, Raghavan 98]
- Automatically finds subspaces with high-density clusters
- Can be considered as both density-based and grid-based
  - Partition the data space S into non-overlapping rectangular units which has the same interval in each dimension
  - Calculate selectivity in each unit, which is a fraction of total data points contained in the unit
  - A unit u is dense if selectivity(u) is grater than threshold
  - Partitioning interval and density threshold are input parameter that user can define

CLIQUE

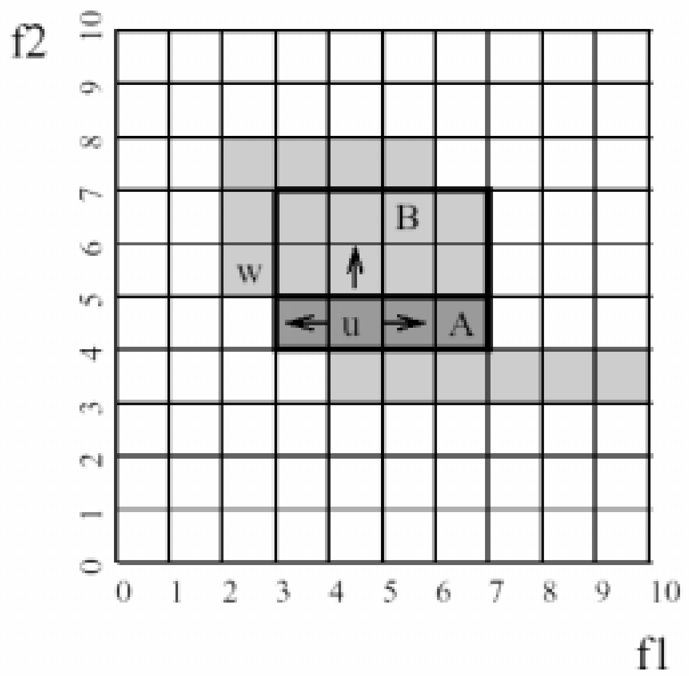




CLIQUE

- Find dense units in bottom-up fashion
  - Use monotonicity : If a set of point S is a cluster in k-dimensional space, then S is also cluster in any (k-1) dimensional projections of this space
  - Having determined (k-1) dimensional dense units, the candidate k dimensional units are determined like Apriori algorithm
- Find cluster
  - After finding dense units, find connected units that would be a cluster
- Generate minimal description for the clusters
  - NP-hard problem
  - Use greedy method

CLIQUE



ROCK

- [Guha, Rastogi, Shim 99]
- Hierarchical clustering algorithm for categorical attributes
  - Example: market basket customers
- Use novel concept of links for merging clusters
  - sim(pi, pj): similarity function that captures the closeness between pi and pj
  - pi and pj are said to be neighbors if sim(pi, pj) $\geq \theta$
  - link(pi, pj): the number of common neighbors
- A new goodness measure was proposed
- Random sampling used for scale up
- Use labeling phase

ROCK

<1, 2, 3, 4, 5>

{1, 2, 3}	{1, 4, 5}
{1, 2, 4}	{2, 3, 4}
{1, 2, 5}	{2, 3, 5}
{1, 3, 4}	{2, 4, 5}
{1, 3, 5}	{3, 4, 5}

<1, 2, 6, 7>

{1, 2, 6}
{1, 2, 7}
{1, 6, 7}
{2, 6, 7}

$$sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|} \geq 0.5$$

- {1, 2, 6} and {1, 2, 7} have 5 links.
- {1, 2, 3} and {1, 2, 6} have 3 links.

Clustering for Categorical Attributes

- Traditional algorithms do not work well for categorical attributes
- Jaccard coefficient has been used for categorical attributes
  - Centroid approach cannot be used
  - Group average and MST algorithms tend to fail
  - Hard to reflect the properties of the neighborhood of the points
  - Fail to capture the natural clustering of data sets
- Viewing as points with (0/1) values of attributes fails too!

Example (Traditional Alg.)

- As the cluster size grows
  - The number of attributes appearing in mean go up
  - Their values in the mean decreases
  - Thus, very difficult to distinguish two points on few attributes

ripple effect

Database: {1, 2, 3, 5} {2, 3, 4, 5} {1, 4} {6}

(0.5,1,1,0.5,1,0) (0.5,0,0,0.5,0,0.5)

(1,1,1,0,1,0) (0,1,1,1,1,0) (1,0,0,1,0,0) (0,0,0,0,0,1)

Conclusions

- CURE and ROCK are interesting algorithms