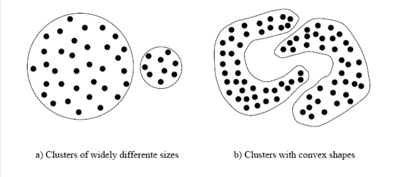
Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling

By George Karypis, Eui-Hong Han,Vipin Kumar

IEEE Computer 32(8): 68-75, 1999

Existing Algorithms

- K-means and PAM
  - Algorithm assigns K-representational points to the clusters and tries to form clusters based on the distance measure.



a) Clusters of widely differente sizes

b) Clusters with convex shapes

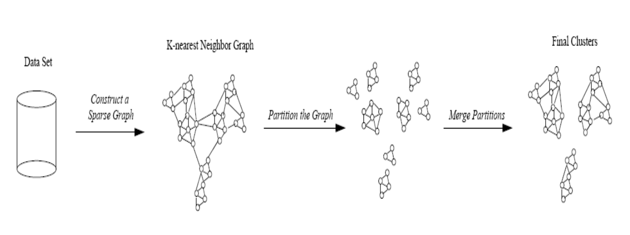
More algorithms

- Other algorithm include CURE, ROCK, CLARANS, etc.
- CURE takes into account distance between representatives
- ROCK takes into account inter-cluster aggregate connectivity.

Chameleon

- Two-phase approach
- Phase -I
  - Uses a graph partitioning algorithm to divide the data set into a set of individual clusters.
- Phase -II
  - uses an agglomerative hierarchical mining algorithm to merge the clusters.

So, basically..



Why not stop with Phase-I? We've got the clusters, haven't we ?

- Chameleon(Phase-II) takes into account
  - Inter Connectivity
  - Relative Closeness

- Hence, chameleon takes into account features intrinsic to a cluster.

Constructing a sparse graph

- Using KNN
  - Data points that are far away are completely avoided by the algorithm (reducing the noise in the dataset)
  - captures the concept of neighbourhood dynamically by taking into account the density of the region.

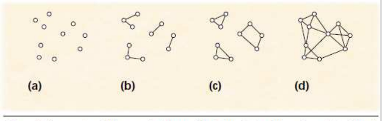
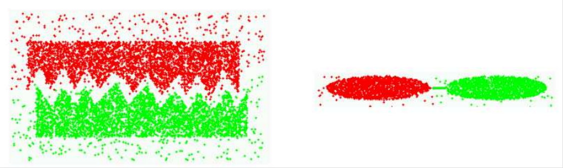


Figure 4. K-nearest-neighbor graphs from original data in two dimensions: (a) original data, (b) 1-, (c) 2-, and (d) 3-nearest neighbor graphs.

What do you do with the graph ?

- Partition the KNN graph such that the edge cut is minimized.
  - Reason: Since edge cut represents similarity between the points, less edge cut => less similarity.
- Multi-level graph partitioning algorithms to partition the graph – hMeTiS library.

Example:

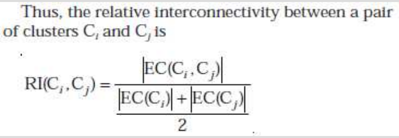


Cluster Similarity

- Models cluster similarity based on the relative inter-connectivity and relative closeness of the clusters.

Relative Inter-Connectivity

- C_i and C_j



Thus, the relative interconnectivity between a pair of clusters C_i and C_j is

$$\mathrm{RI}(C_i, C_j) = \frac{|\mathrm{EC}(C_i, C_j)|}{\frac{|\mathrm{EC}(C_i)| + |\mathrm{EC}(C_j)|}{2}}.$$

  - where $EC(C_i,C_j)$= sum of weights of edges that connect C_i with C_j.
  - $EC(C_i)$ = weighted sum of edges that partition the cluster into roughly equal parts.

Relative Closeness

- Absolute closeness normalized with respect to the internal closeness of the two clusters.
- Absolute closeness got by average similarity between the points in C_i that are connected to the points in C_j.

average weight of the edges from C_i->C_j.

Internal Closeness….

- Internal closeness of the cluster got by average of the weights of the edges in the cluster.

$$RC(C_i, C_j) = \frac{\overline{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\overline{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\overline{S}_{EC_{C_j}}},$$

So, which clusters would we like to merge?

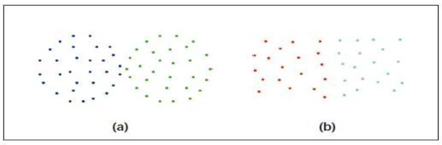


Figure 1. Algorithms based only on closeness will incorrectly merge (a) the dark-blue and green clusters because these two clusters are closer together than (b) the red and cyan clusters.

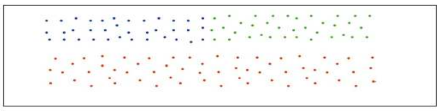


Figure 2. Algorithms based only on the interconnectivity of two clusters will incorrectly merge the dark-blue and red clusters rather than dark-blue and green clusters.

So, which clusters do we merge?

- So far, we have got
  - Relative Inter-Connectivity measure.
  - Relative Closeness measure.
- Using them,

$$RI(C_i, C_j) * RC(C_i, C_j)^{\alpha},$$

Experimental Results

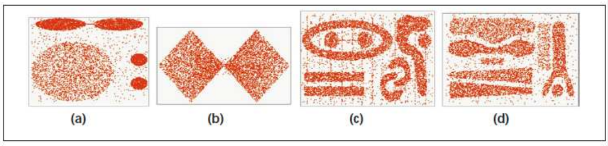


Figure 5. Four data sets used in our experiments: (a) DS1 has 8,000 points; (b) DS2 has 6,000 points; (c) DS3 has 10,000 points; and (d) DS4 has 8,000 points.

Experimental Results..

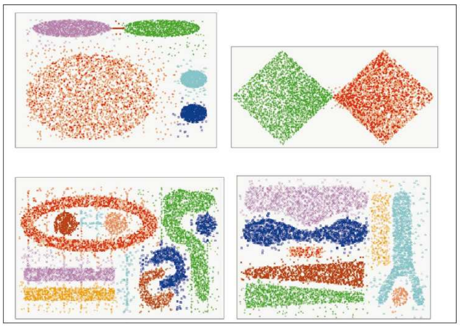


Figure 6. Clusters discovered by Chameleon for the four data sets.

Experimental Results

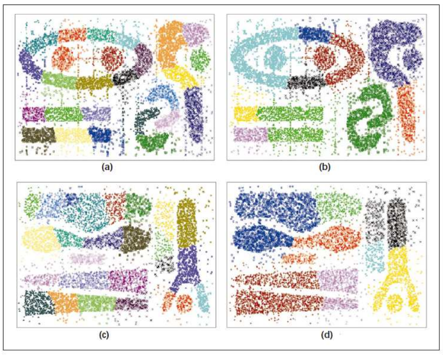
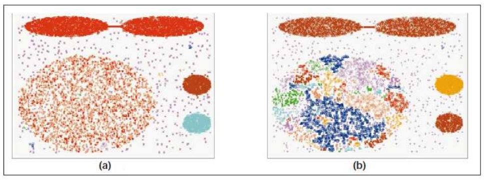


Figure 7. Clusters identified by CURE with shrinking factor 0.3 and number of representative points equal to 10. For DS3, CURE first merges subclusters that belong to two different subclusters at (a) 25 clusters. With (b) 11 clusters specified—the same number that Chameleon found—CURE obtains the result shown. CURE produced these results for DS4 with (c) 25 and (d) 8 clusters specified.

Experimental Results

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.



Experimental Results

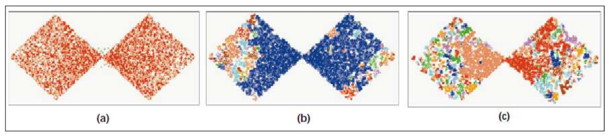


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.

Good points about the paper :

- Nice description of the working of the system.
- Gives a note of existing algorithms and as to why chameleon is better.
- Not specific to a particular domain.

yucky and reasonably yucky parts..

- Not much information given about the Phase-I part of the paper – graph properties?
- Finding the complexity of the algorithm
  O(nm + n log n + m^2log m)
- Different domains require different measures for connectivity and closeness, ...

Questions ?

AS YOU CAN CLEARLY SEE IN SLIDE 397...
GAAAAH!
"POWERPOINT" POISONING.
www.dilbert.com scottadams@aol.com
© 2000 United Feature Syndicate, Inc.