

CMU SCS

15-826: Multimedia Databases and Data Mining

Data Mining - clustering (revisited)
C. Faloutsos

CMU SCS

Outline

Goal: 'Find **similar / interesting** things'

- Intro to DB
- Indexing - similarity search
- ➔ • Data Mining

15-826 Copyright: C. Faloutsos (2006) 2

CMU SCS

Data Mining - Detailed outline

- Statistics
- AI - decision trees
- DB
 - data warehouses; data cubes; OLAP
 - classifiers
 - association rules
 - misc. topics:
 - clustering (revisited)
 - reconstruction of info

➔

15-826 Copyright: C. Faloutsos (2006) 3

CMU SCS

Clustering - outline

- ➔ • Partitioning methods (eg., k-means)
- Hierarchical methods (eg., agglomerative)
- Density-based methods
- Grid-based methods

15-826 Copyright: C. Faloutsos (2006) 4

CMU SCS

Partitioning methods

K-means (k is given). Goal:

- find k-points and assign the nearest data points to them
- to minimize the sum of squared distances

Algorithm: Iterative improvement
Can be expensive (needs many iterations)
Sensitive to outliers

15-826 Copyright: C. Faloutsos (2006) 5


CMU SCS

Partitioning methods

Variations:

- k-medoids / CLARANS (but: $O(n^2)$)
- k harmonic means [Zhang, Hsu, Dayal+ HPLabs, 99]

15-826 Copyright: C. Faloutsos (2006) 6




CMU SCS

Clustering - outline

- Partitioning methods (eg., k-means)
- ➔ Hierarchical methods (eg., agglomerative)
- Density-based methods
- Grid-based methods

15-826 Copyright: C. Faloutsos (2006) 7




CMU SCS

Hierarchical methods

- Agglomerative / Divisive Hierarchical Clustering
- BIRCH
- CURE
- CHAMELEON

15-826 Copyright: C. Faloutsos (2006) 8




CMU SCS

Hierarchical methods

BIRCH:

- uses the 'CF'-tree (count, center, sum of squares, for the points in each node)
- needs 1 or 2 passes
- finds spherical clusters
- needs two numbers: branching factor; radius threshold

15-826 Copyright: C. Faloutsos (2006) 9



CMU SCS

Hierarchical methods


BIRCH Algorithm - phase 1:

- start inserting points in the CF-tree (~ a sphere tree)
- if the radius, is exceeded, split the node

phase 2:

- cluster the leaf nodes of the tree

15-826 Copyright: C. Faloutsos (2006) 10




CMU SCS

CURE

Main ideas:

- use sampling
- represent a cluster by many centroids (thus clusters can have arbitrary shapes)

15-826 Copyright: C. Faloutsos (2006) 11



CMU SCS

CURE

Algorithm:

- get a sample
- partition it into a set of partitions
- partially cluster each partition
- cluster the partial clusters

15-826 Copyright: C. Faloutsos (2006) 12

CMU SCS

CURE

sample partition partially cluster

15-826 Copyright: C. Faloutsos (2006) 13

CMU SCS

CURE

group clusters

- $O(n)$ complexity
- good quality clusters
- relatively small sensitivity to user parameters

15-826 Copyright: C. Faloutsos (2006) 14

CMU SCS

CHAMELEON

- for each object, link it to its k-nn
- use graph partitioning, to create many small clusters (= connected components)
- merge clusters that are 'close enough'

15-826 Copyright: C. Faloutsos (2006) 15

CMU SCS

CHAMELEON

- similarity of a pair of clusters C_i, C_j : use
 - 'relative inter-connectivity' (= avg # of cross-links, normalized)
 - 'relative closeness' (= avg pairwise distance, normalized)
- Empirically: better quality clusters, but $O(n^2)$

15-826 Copyright: C. Faloutsos (2006) 16

CMU SCS

Clustering - outline

- Partitioning methods (eg., k-means)
- Hierarchical methods (eg., agglomerative)
- ➡ Density-based methods
- Grid-based methods

15-826 Copyright: C. Faloutsos (2006) 17

CMU SCS

DBSCAN

- High level idea: group high-density areas, that are within a threshold
- in detail: decide on
 - ϵ : a distance threshold and
 - minPts: minimum number of points in a neighborhood
- 'core object': iff it has \geq minPts within ϵ

15-826 Copyright: C. Faloutsos (2006) 18

DBSCAN

- Pictorially (say, minPts=3):

The diagram shows a set of points in a 2D space. A cluster of four points is circled, and an arrow points to it with the label "core object". A horizontal line with an arrow labeled ϵ indicates the radius of the neighborhood around the core object.

15-826 Copyright: C. Faloutsos (2006) 19

DBSCAN

- Pictorially (say, minPts=3):

- 'directly density reachable'
- 'density-connected'

The diagram shows the same set of points as slide 19. Two overlapping circles represent the ϵ neighborhoods of two different core objects. An arrow points to one of the core objects with the label "core object".

15-826 Copyright: C. Faloutsos (2006) 20

DBSCAN

- can give elongated clusters
- needs $O(n*n)$ time at worst; less, with a spatial index (but: dim. curse...)
- Also, sensitive to ϵ (which is global = not adaptive)
- hence: OPTICS

15-826 Copyright: C. Faloutsos (2006) 21

OPTICS

- Given 'minPts'
- impose a sequential ordering on the points
- estimate the 'reachability distance' and
- look for plateaus

(Details are tricky - intuitively, somehow similar to traversing an MST, and plotting the edge length for each node)

15-826 Copyright: C. Faloutsos (2006) 22

OPTICS

The diagram shows a scatter plot of points in the top right. Below it is a graph with "reachability distance" on the y-axis and "rank of object" on the x-axis. The graph shows a curve that starts high, drops to a low plateau, rises to a sharp peak, and then drops again to a low plateau.

15-826 Copyright: C. Faloutsos (2006) 23

OPTICS

The diagram shows the same scatter plot and reachability distance graph as slide 23. Two peaks in the graph are highlighted with green and red arrows. Above the scatter plot, two overlapping circles (one green, one red) represent the ϵ neighborhoods of the points corresponding to the highlighted peaks.

15-826 Copyright: C. Faloutsos (2006) 24

CMU SCS

OPTICS

- speed: like DBSCAN ($O(n^2)$), or less, with a spatial index

15-826 Copyright: C. Faloutsos (2006) 25

CMU SCS

Clustering - outline

- Partitioning methods (eg., k-means)
- Hierarchical methods (eg., agglomerative)
- Density-based methods
- Grid-based methods

15-826 Copyright: C. Faloutsos (2006) 26

CMU SCS

Grid-based methods

- Main idea: impose a grid; group together 'similar' nearby cells
- STING: do a quad-tree decomposition. For each cell:
 - keep statistics,
 - test (chi-square) whether the distribution is known (uniform, Gauss, etc)
 - merge similar cells together and
 - repeat recursively


15-826 Copyright: C. Faloutsos (2006) 27

CMU SCS

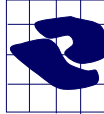
Grid-based methods

'WaveCluster' (2- or 3-d address space)

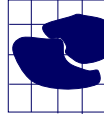
- do wavelet transform first
- create a hierarchy of clusters, one for each resolution, by grouping the connected components



hi-res



med-res



low-res

15-826 Copyright: C. Faloutsos (2006) 28

CMU SCS

Grid-based methods

'WaveCluster' : Fast ($O(n)$), but only for low-dimensionality

15-826 Copyright: C. Faloutsos (2006) 29

CMU SCS

Grid-based methods

'CLIQUE': for high dimensions
 'Dense' cell : has $\geq k$ points
 Goal: find 'dense' cells, in m , or lower, dimensions

15-826 Copyright: C. Faloutsos (2006) 30

CMU SCS

Grid-based methods

Idea:

- project in lower dimensionalities;
- report dense cells, that are connected
- look for higher-dimensionality dense cells

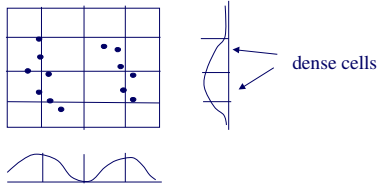
Uses an 'a-priori'-like argument:
if a cell is 'dense', so are all its projections

15-826 Copyright: C. Faloutsos (2006) 31

CMU SCS

Grid-based methods

Pictorially:

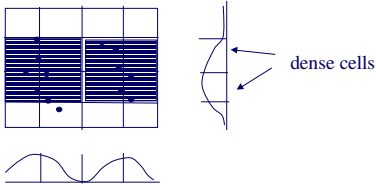


15-826 Copyright: C. Faloutsos (2006) 32

CMU SCS

Grid-based methods

Pictorially:



15-826 Copyright: C. Faloutsos (2006) 33

CMU SCS

Clustering - outline

- Partitioning methods (eg., k-means)
- Hierarchical methods (eg., agglomerative)
- Density-based methods
- Grid-based methods
- ➔ Conclusions

15-826 Copyright: C. Faloutsos (2006) 34

CMU SCS

Conclusions

- $O(n)$ methods: BIRCH, CURE, CLIQUE
- in between: DBSCAN, OPTICS
- $O(n^2)$: CHAMELEON

15-826 Copyright: C. Faloutsos (2006) 35

CMU SCS

Reference

- Han + Kamber, chapter 8.4-8.7

15-826 Copyright: C. Faloutsos (2006) 36