DBSCAN

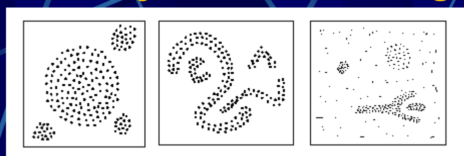
Outline

- Clustering Background
- Density-based Clustering
- DBSCAN Algorithm
- DBSCAN Implementation on ATLaS
- Performance
- Conclusion

Clustering Algorithms

- Partitioning Alg: Construct various partitions then evaluate them by some criterion (CLARANS, O(n) calls)
- Hierarchy Alg: Create a hierarchical decomposition of the set of data (or objects) using some criterion (merge & divisive, difficult to find termination condition)
- Density-based Alg: based on local connectivity and density functions

Density-Based Clustering



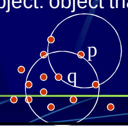
- Clustering based on density (local cluster criterion), such as density-connected points
- Each cluster has a considerable higher density of points than outside of the cluster

Density-Based Clustering

- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - GDBSCAN: Sander, et al. (KDD'98)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

Density Concepts

- Two global parameters:
  - ***Eps***: Maximum radius of the neighbourhood
  - ***MinPts***: Minimum number of points in an Eps-neighbourhood of that point
- Core Object: object with at least MinPts objects within a radius 'Eps-neighborhood'
- Border Object: object that on the border of a cluster



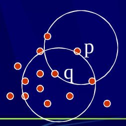
MinPts = 5

Eps = 1 cm

Density-Based Clustering: Background

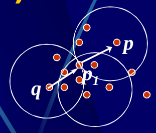
- $N_{Eps}(p)$: {q belongs to D | dist(p,q) <= Eps}
- Directly density-reachable: A point ***p*** is directly density-reachable from a point ***q*** wrt. ***Eps***, ***MinPts*** if
  - 1) ***p*** belongs to $N_{Eps}(q)$
  - 2) $|N_{Eps}(q)|$ >= ***MinPts***

    (core point condition)

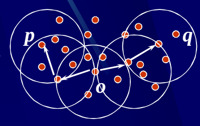


Density-Based Clustering: Background (II)

- Density-reachable:
  - A point *p* is density-reachable from a point *q* wrt. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

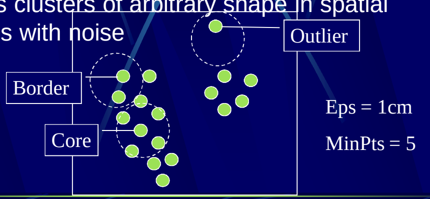


- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



DBSCAN: Density Based Spatial Clustering of Applications with Noise

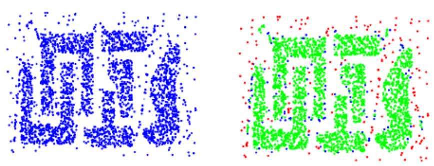
- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

- Arbitrary select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

DBSCAN: Core, Border and Noise Points



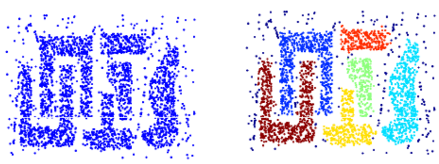
Original Points

Point types: core, border and noise

Eps = 10, MinPts = 4

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 52

When DBSCAN Works Well



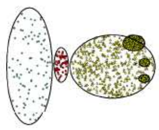
Original Points

Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

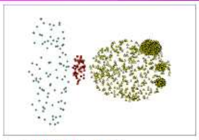
© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 53

When DBSCAN Does NOT Work Well

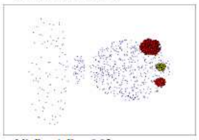


Original Points

- Varying densities
- High-dimensional data



(MinPts=4, Eps=9.75).

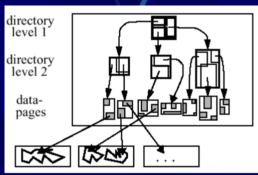


(MinPts=4, Eps=9.92)

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 54

R*-Tree(1)

- R*-Tree: A spatial index
- Generalize the 1-dimensional B+Tree to d-dimensional data spaces



R*-tree(2)

- R*-Tree is a height-balanced data structure
- Search key is a collection of d-dimensional intervals
- Search key value is referred to as bounding boxes

R*-Tree(3)

- Query a bounding box B in R*-Tree:
  - Test bounding box for each child of root
  - if it overlaps B, search the child's subtree
  - If more than one child of root has a bounding box overlapping B, we must search all the corresponding subtrees
  - Important difference between B+tree: search for single point can lead to several paths

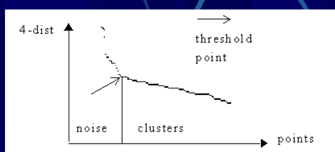
DBSCAN Complexity Comparison

Time Complexity	A single neighborhood query	DBSCAN
Without index	O(n)	O(n^2)
R*-tree	O(log n)	O(n*log n)

- The height of a R*-Tree is O(log n) in the worst case
- A query with a "small" region traverses only a limited number of paths in the R*-Tree
- For each point, at most one neighborhood query is needed

Heuristic for Eps and Minpts

- K-dist (p): distance from the kth nearest neighbour to p
- Sorting by k-dist (p)

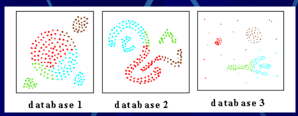


- Minpts: k>4 no significant difference, but more computation, thus set k=4

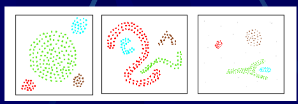
Performance Evaluation compared with CLARANS (1)

- Accuracy

CLARANS:



DBSCAN:



Performance Evaluation compared with CLARANS (2)

- Efficiency

SEQUOIA2000 benchmark data (Stonebraker et al. 1993)

number of points	1252	2503	3910	5213	6256
DBSCAN	3.1	6.7	11.3	16.0	17.8
CLAR-ANS	758	3026	6845	11745	18029
number of points	7820	8937	10426	12512	
DBSCAN	24.5	28.2	32.7	41.7	
CLAR-ANS	29826	39265	60540	80638	

Advantages

- DBSCAN does not require you to know the number of clusters in the data a priori, as opposed to k-means.
- DBSCAN can find arbitrarily shaped clusters. It can even find clusters completely surrounded by (but not connected to) a different cluster. Due to the MinPts parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced.
- DBSCAN has a notion of noise.

Advantages

- DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database. (Only points sitting on the edge of two different clusters might swap cluster membership if the ordering of the points is changed, and the cluster assignment is unique only up to isomorphism.)

Disadvantages

- DBSCAN can only result in a good clustering as good as its distance measure is in the function getNeighbors(P,epsilon). The most common distance metric used is the euclidean distance measure. Especially for high-dimensional data, this distance metric can be rendered almost useless due to the so called "Curse of dimensionality", rendering it hard to find an appropriate value for epsilon. This effect however is present also in any other algorithm based on the euclidean distance.
- DBSCAN cannot cluster data sets well with large differences in densities, since the minPts-epsilon combination cannot be chosen appropriately for all clusters then.

References

- Ester M., Kriegel H.-P., Sander J. and Xu X. 1996. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining. Portland, OR, 226-231.
- Raghu Ramakrishnan, Johannes Gehrke, “Database Management systems (Second Edition)”, McGraw-Hill Companies, Inc.
- Beckmann N., Kriegel H.-P., Schneider R, and Seeger B. 1990. “The R*-tree: An Efficient and RobustAccess Method for Points and Rectangles”. Proc. ACM SIGMOD Int. Conf. on Management of Data.Atlantic City, NJ, 322-331.
- Jain A.K., and Dubes R.C. 1988. “Algorithms for Clustering Data”. New Jersey: Prentice Hall.
- Sander J., Ester M., Kriegel H.-P., Xu X.: Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications, in: Data Mining and Knowledge Discovery, an Int. Journal, Kluwer Academic Publishers, Vol. 2, No. 2, 1998, pp. 169-194.
- Haixun Wang, Carlo Zaniolo: Database System Extensions for Decision Support: the AXL Approach. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery 2000: 11-20