Data Mining
Cluster Analysis: Basic Concepts and Algorithms

Lecture Notes for Chapter 5

Data Mining

by

Zhaonian Zou

5.4 Density-based Clustering

5.4.1 What is Density-based Clustering?

What is Density-based Clustering?

- Density-based clustering locates regions of high density that are separated from one another by regions of low density.

Density-based Clustering Algorithms

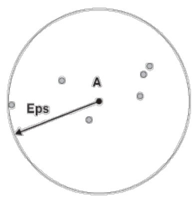
- DBSCAN Algorithm
- OPTICS Algorithm
- Grid-based Algorithm
- CLIQUE Algorithm
- DENCLUE Algorithm

5.4 Density-based Clustering

5.4.2 DBSCAN Algorithm

Center-based Density

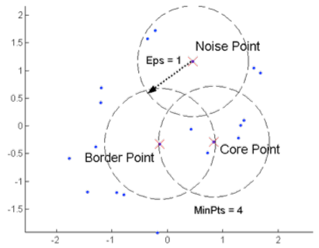
- Density = number of points within a specified radius (Eps) of a point



Density = 7 (including A itself)

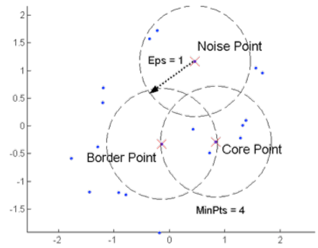
Core Points

- A point is a core point if it has more than a specified number of points (MinPts) within a radius Eps
  - A core point is in the interior of a cluster



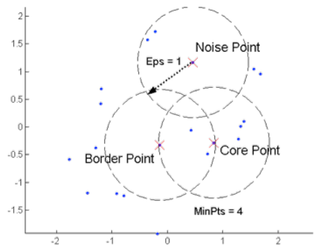
Border Points

- A point is a border point if it has fewer than MinPts points within a radius Eps, but is in the neighborhood of a core point
  - A border point is on the border of a cluster

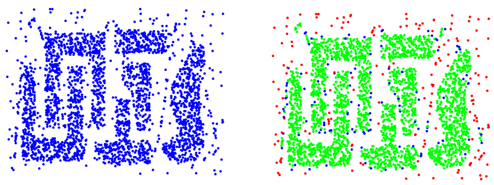


Noise Points

- A noise point is any point that is neither a core point nor a border point.
  - A noise point is in a sparely occupied region



Core, Border and Noise Points



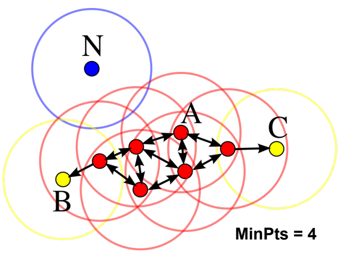
Original Points

Point types: core, border and noise

Eps = 10, MinPts = 4

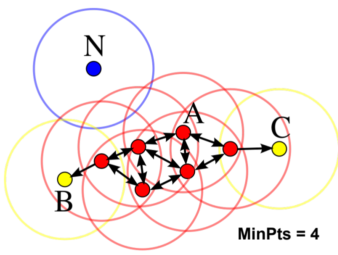
Density-based Clusters in DBSCAN

- Any two core points that are within a distance Eps of one another are in the same cluster



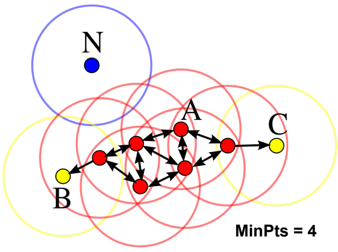
Density-based Clusters in DBSCAN

- Any border point that is within a radius Eps of a core point is put in the same cluster as the core point (ties are broken arbitrarily)



Density-based Clusters in DBSCAN

- All noise points are discarded



DBSCAN Algorithm

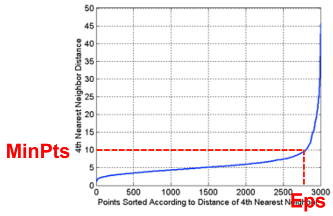
- Eliminate noise points
- Perform clustering on the remaining points
  - Put an edge between all core points that are within Eps of each other
  - Make each connected component as a separate cluster
  - Assign each border point to one of the clusters of its associated core points

Time Complexity

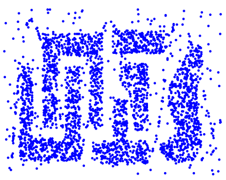
- The original KDD'96 paper misclaimed O(nlogn) running time, where n is the number of objects.
- Gan and Tao's SIGMOD'15 Best Paper
  - DBSCAN actually requires $O(n^2)$ time.
  - The running time can be dramatically brought down to O(n) in expectation as soon as slight inaccuracy in the clustering results is permitted.
  - http://www.cse.cuhk.edu.hk/~taoyf/paper/sigmod15-dbscan.pdf

Determining EPS and MinPts

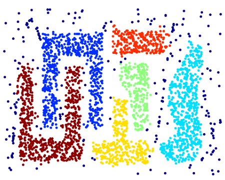
- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor



When DBSCAN Works Well



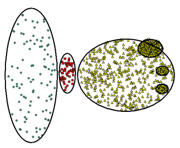
Original Points



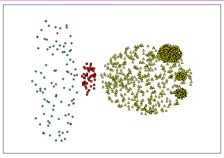
Clusters

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

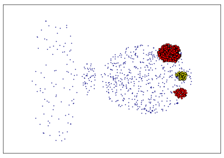
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

- Varying densities
- High-dimensional data

5.4 Density-based Clustering

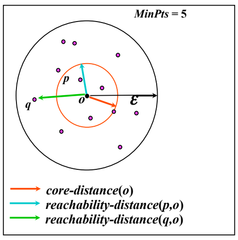
5.4.3 OPTICS Algorithm

OPTICS

- OPTICS = Ordering Points to Identify Clustering Structure
- OPTICS addresses one of DBSCAN's major weaknesses: the problem of detecting meaningful clusters of varying density
- OPTICS (linearly) orderes the points so that points which are spatially closest become neighbors in the ordering

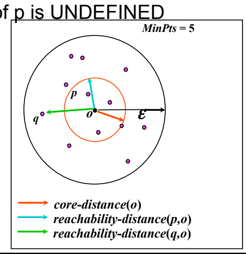
Core Distance

- The core distance of a point is its distance to the MinPts-th closest point
  - The core distance is UNDEFINED if the distance to the MinPts-th closest point is larger than Eps



Reachability Distance

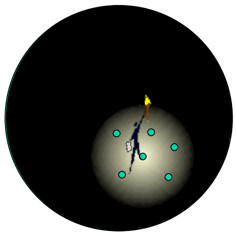
- The reachability distance of another point p from a point o is the larger one of the distance between p and o and the core distance of o
  - The reachability distance is UNDEFINED if the core distance of p is UNDEFINED



OPTICS Algorithm: Basic Idea

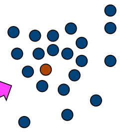
OPTICS Algorithm: Basic Idea

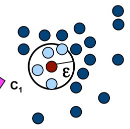
- Order points by shortest reachability distance with respect to the points seen so far to guarantee that clusters w.r.t. higher density are finished first
- Visit each point by making always a shortest jump

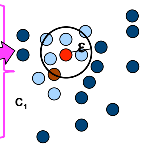


OPTICS Algorithm

```
for each point p of DB
    p.reachability-distance = UNDEFINED
for each unprocessed point p of DB
    N = getNeighbors(p, eps)
    mark p as processed
    output p to the ordered list
    if (core-distance(p, eps, Minpts) != UNDEFINED)
        Seeds = empty priority queue
        update(N, p, Seeds, eps, Minpts)
        for each next q in Seeds
            N' = getNeighbors(q, eps)
            mark q as processed
            output q to the ordered list
            if (core-distance(q, eps, Minpts) != UNDEFINED)
                update(N', q, Seeds, eps, Minpts)
```

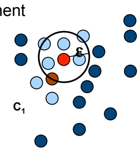






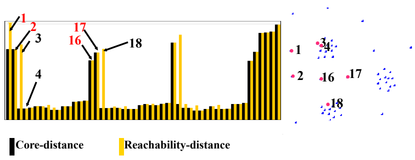
OPTICS Algorithm ...

```
update(N, p, Seeds, eps, Minpts)
    coredist = core-distance(p, eps, MinPts)
    for each o in N
        if (o is not processed)
            new-reach-dist = max(coredist, dist(p,o))
            if (o.reachability-distance == UNDEFINED) // o is not in Seeds
                o.reachability-distance = new-reach-dist
                Seeds.insert(o, new-reach-dist)
            else            // o in Seeds, check for improvement
                if (new-reach-dist < o.reachability-distance)
                    o.reachability-distance = new-reach-dist
                    Seeds.move-up(o, new-reach-dist)
```



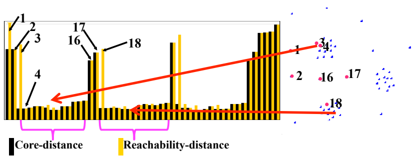
Identifying Noise Points

- A noise point o with respect to Eps* < Eps and MinPts satisfies core-distance(o) > Eps* and reachablity-dist(o) > Eps*



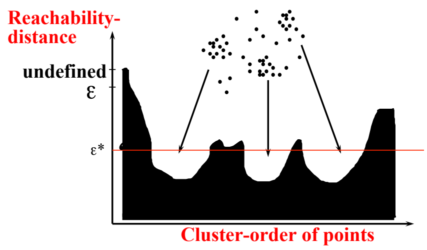
Extracting Clusters

- A cluster with respect to Eps* < Eps and MinPts
  - Starts with an object o where core-distance(o) <= Eps* and reachablity-dist(o) > Eps*
  - Continues while reachability-dist <= Eps*



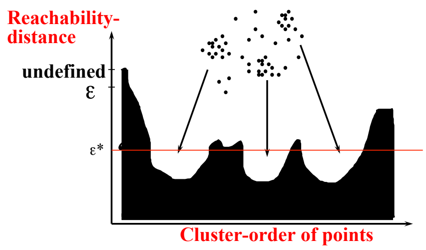
Reachability Plot

- The ordering of the points on the x-axis is as processed by OPTICS
- Y-axis is the reachability distance of points



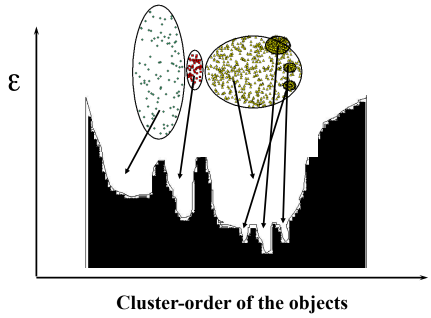
Reachability Plot ...

- The clusters show up as valleys in the reachability plot
  - The deeper the valley, the denser the cluster



Strengths of OPTICS

- OPTICS can detect clusters of varying density

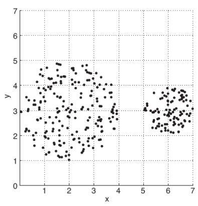


5.4 Density-based Clustering

5.4.4 Grid-based Clustering

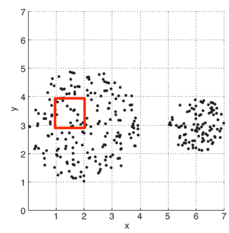
Grid

- Split each attribute into a number of contiguous intervals.
- The grid is composed by a set of cells, each of which is defined by the Cartesian Product the intervals, one from each attribute.



Grid-based Density

- The density of a grid cell is the number of points in the cell divided by the volume of the cell.



0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

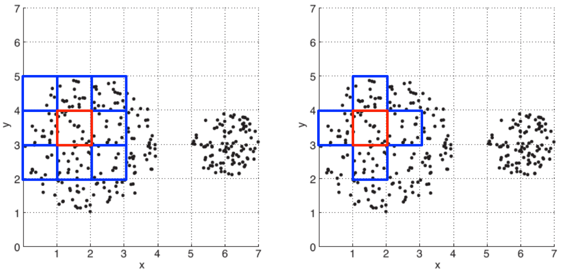
Grid-based Clustering

- Define a set of grid cells.
- Assign objects to cells and compute the density of each cell.
- Eliminate cells having a density below a specified threshold t.
- Form clusters from contiguous groups of dense cells.

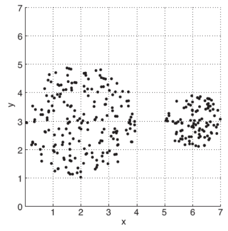
Defining Grid Cells

- The definition of the grid has a strong impact on the clustering results.
- Common Approaches
  - Equal-width binning
  - Equal-height binning
  - Hierarchical clustering

Definition of Contiguous Cells



Grid-based Algorithm



0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

Limitations of Grid-based Clustering

- Choosing the right threshold t is difficult.
  - If t is too high, some clusters may be lost.
  - If t is too small, some separate clusters may be joined.
- Inaccurately handle the boundary areas of a globular cluster.
- A group of points may be split into two cells, which are discarded later.
- Grid-based clustering tends to work poorly for high-dimensional data. (Curse of dimensionality!)

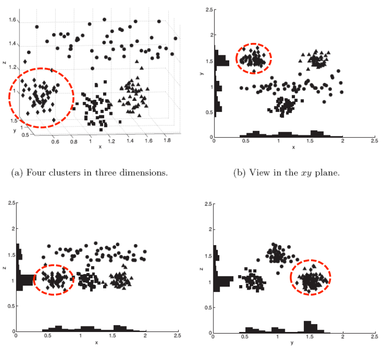
5.4 Density-based Clustering

5.4.5 Subspace Clustering

Subspace Clusters: an Illustration

This set of points is a cluster in 3D space.

This set of points is also part of a cluster in each 2D subspace.



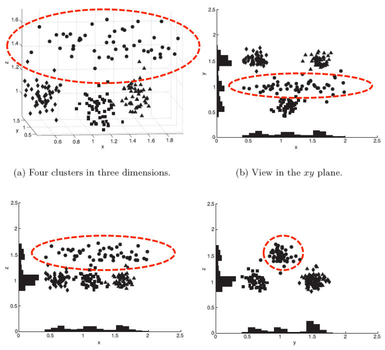
(a) Four clusters in three dimensions. (b) View in the xy plane.

(c) View in the xy plane. (d) View in the xy plane.

Subspace Clusters: an Illustration

This set of points is not a cluster in 3D space.

This set of points forms a cluster in each 2D subspace.



(a) Four clusters in three dimensions. (b) View in the xy plane.

(c) View in the xy plane. (d) View in the xy plane.

Why Subspace Clustering?

- The data may be clustered with respect to a subset of attributes, but randomly distributed with respect to the remaining attributes.
- It is possible that different clusters exist in different subspaces.
- **Challenges:** The number of subspaces is exponential in the number of dimensions!

Subspace Clustering

- **Input:** a set of points in d-dimensional space
- **Output:** the clusters and the dimensions in which they exist
- **Constraints:** only output the clusters that are not projections of higher-dimensional clusters.

CLIQUE Algorithm

- The CLIQUE algorithm is similar to the Apriori algorithm.
- CLIQUE relies on the following property:
  - If a set of points forms a density-based cluster with respect to k attributes, then the same set of points is also part of a density-based cluster in all possible subsets of the attributes.

CLIQUE Algorithm

- Find all the dense intervals in the 1D space corresponding to each attribute
- K = 2
- Repeat until there are no candidate dense k-dimensional cells
  - Generate all candidate dense k-dimensional cells from dense (k – 1)-dimensional cells
  - Eliminate cells having a density less than t
  - K = K + 1
- Find clusters by taking the union of all contiguous dense cells.

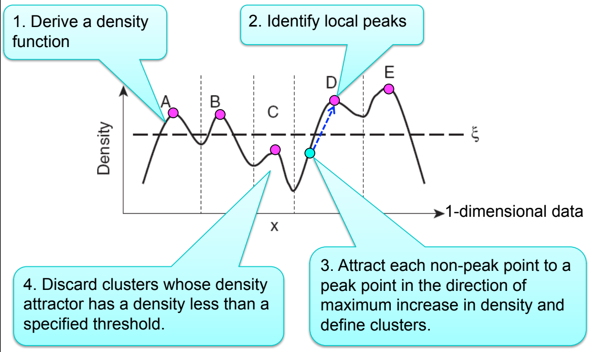
Limitations of CLIQUE

- Limitations of grid-based clustering
- Exponential time complexity

5.4 Density-based Clustering

5.4.6 Kernel-based Clustering

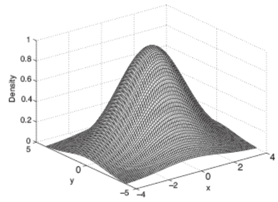
Illustrating Kernel-based Clustering



Kernel Functions

- The kernel function models the influence of each point to the overall density function.
- Gaussian Kernel: the influence of a point y with respect to a certain point x is given by

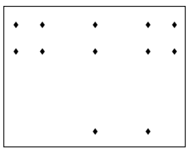
$$K_{\mathbf{x}}(\mathbf{y}) = \exp\left(-\frac{d^2(\mathbf{x},\mathbf{y})}{2\sigma^2}\right)$$



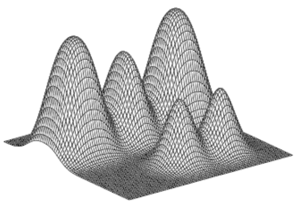
Overall Density Function

- A density function for the d-dimensional space where the data resides is produced by applying the Gaussian kernel function to the set of points.

$$f(\mathbf{x}) = \sum_{\mathbf{y} \in D} K_{\mathbf{x}}(\mathbf{y})$$



Data



Density function