DBSCAN

Presented by:

Garrett Poppe

A density-based algorithm for discovering clusters in large spatial databases with noise

by Martin Ester, Hans-peter Kriegel,

Jörg S, Xiaowei Xu

Slides adapted from resources outlined in the resources slide

Summary

- K-Means Clustering Method
- Density Based Clustering
- DBSCAN
  - Points
  - Optimal Eps & MinPts
  - Algorithm
  - Flaws
  - Complexity
- Resources
- Questions

The *K-Means* Clustering Method: **for numerical attributes**

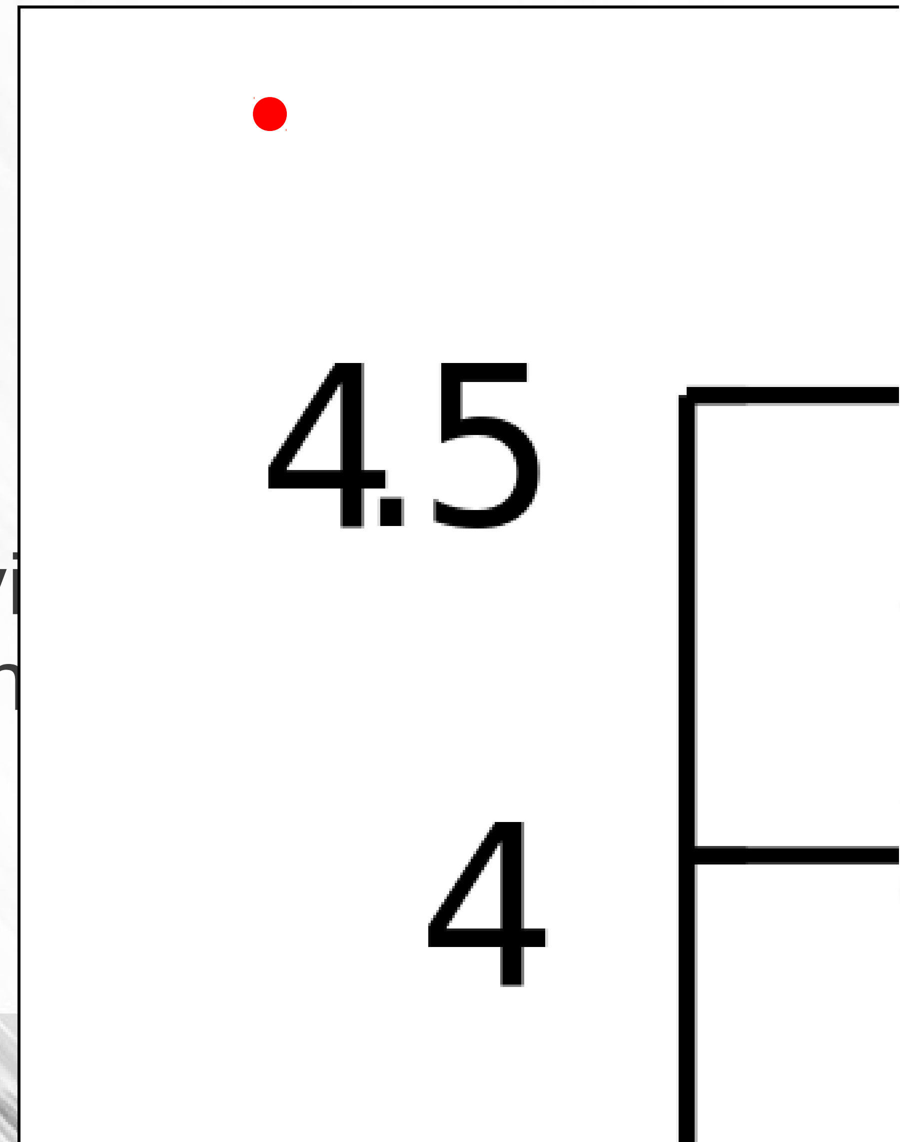
Given *k*, the *k-means* algorithm is implemented in four steps:

- Partition objects into *k* non-empty subsets
- Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
- Assign each object to the cluster with the nearest seed point
- Go back to Step 2, stop when no more new assignment

The *K-Means* Clustering Method

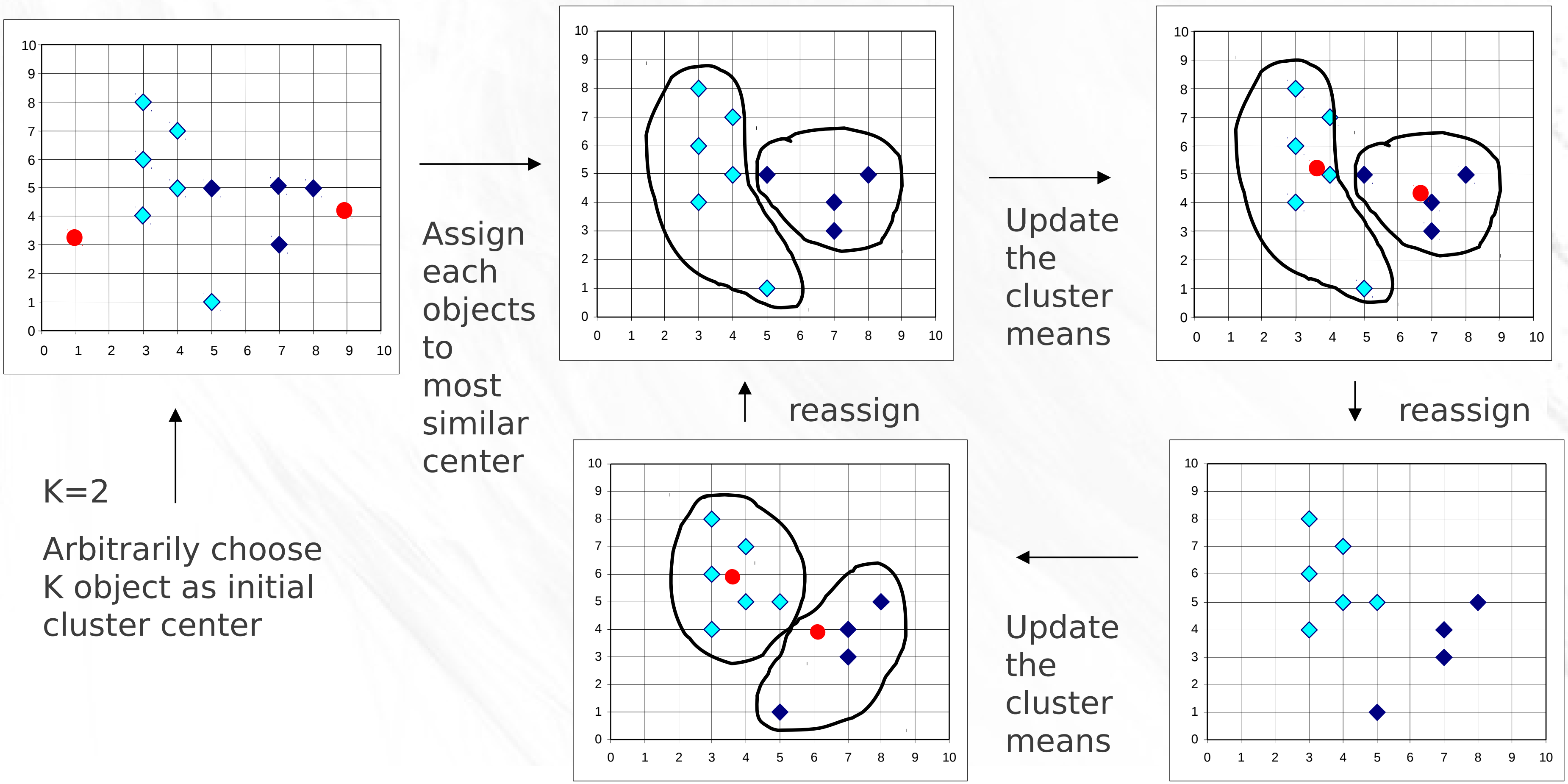
X	Y
1	2
2	4
3	3
4	2
2.5	2.75

The mean point can be a vi
mean point can be influen

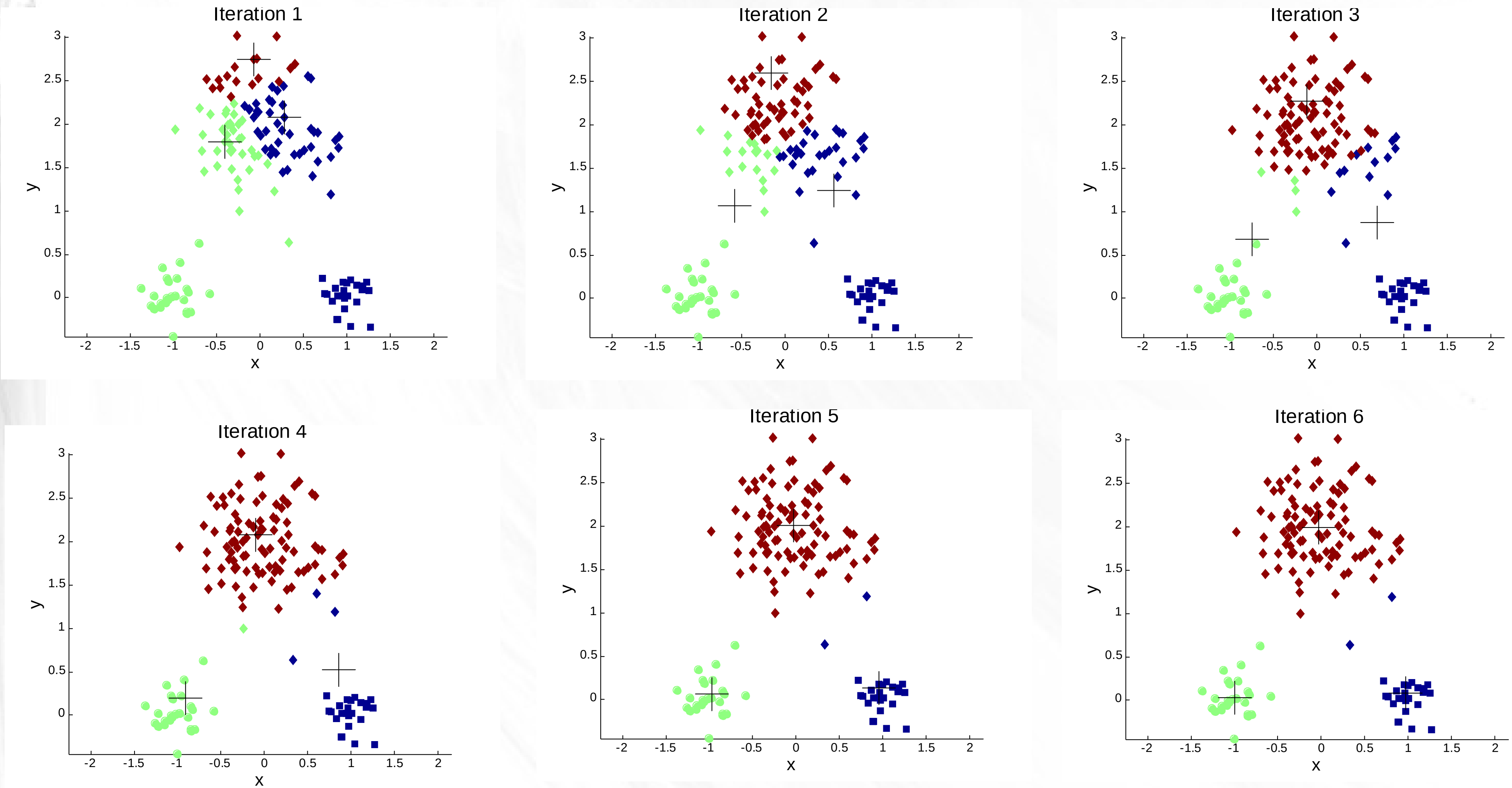


The *K-Means* Clustering Method

- Example



The *K-Means* Clustering Method

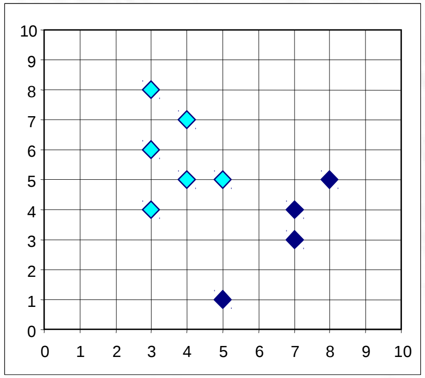


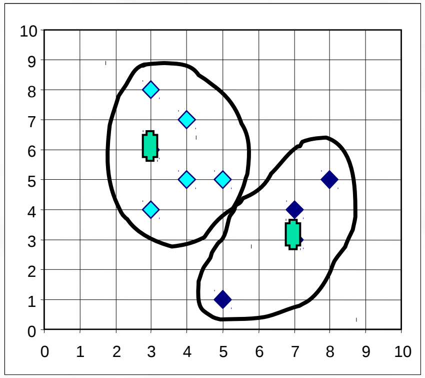
The *K-Means* Clustering Method

The k-means algorithm is sensitive to outliers

Since an object with an extremely large value may substantially distort the distribution of the data.

K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.





Density-Based Clustering Methods

Clustering based on density (local cluster criterion), such as density-connected points

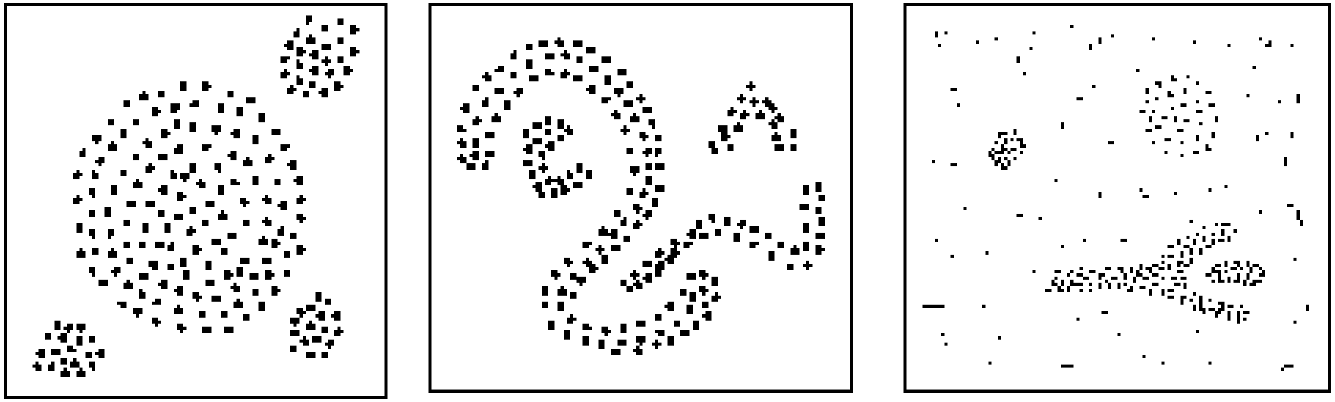
<u>Major features:</u>

- Discover clusters of arbitrary shape
- Handle noise
- One scan
- Need density parameters as termination condition

Several interesting studies:

- <u>DBSCAN:</u> Ester, et al. (KDD'96)
- <u>OPTICS</u>: Ankerst, et al (SIGMOD'99).
- <u>DENCLUE</u>: Hinneburg & D. Keim (KDD'98)
- <u>CLIQUE</u>: Agrawal, et al. (SIGMOD'98)

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

DBSCAN

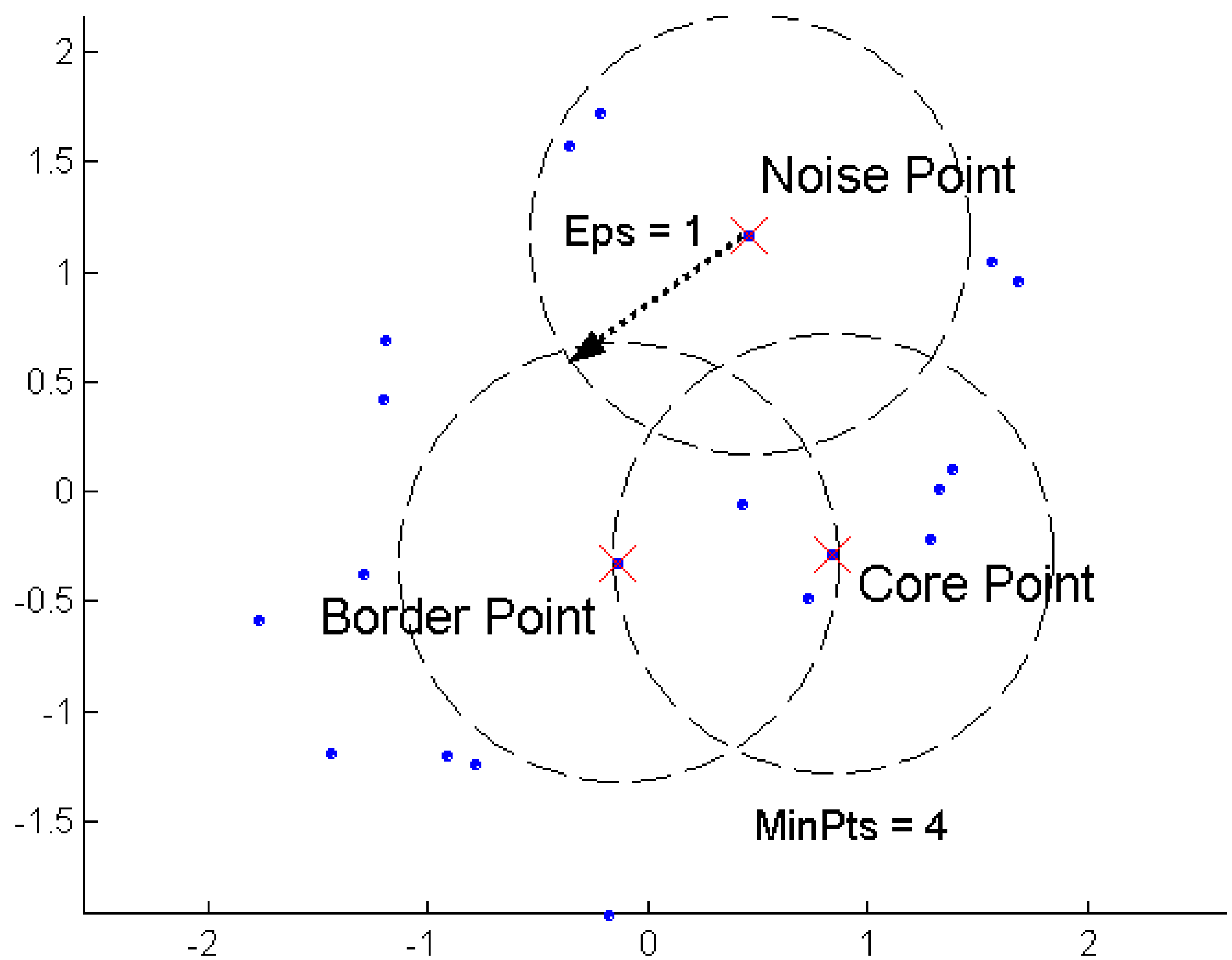
DBSCAN is a density-based algorithm.

- Density = number of points within a specified radius r (Eps)
- A point is a core point if it has more than a specified number of points (MinPts) within Eps

These are points that are at the interior of a cluster

- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
- A noise point is any point that is not a core point or a border point.

DBSCAN:Core, Border, and Noise points



DBSCAN

Two parameters (eps and MinPts)***:***

- ***ε***: Maximum radius of the neighbourhood
- ***MinPts***: Minimum number of points in an Eps-neighbourhood of that point
- ***N_ε(p)***: ***{q belongs to D | dist(p,q) <= ε}***

Directly density-reachable**:** A point ***p*** is directly density-reachable from a point ***q*** wrt. ***ε***, ***MinPts*** if

1) ***p*** belongs to ***N_ε(q)***

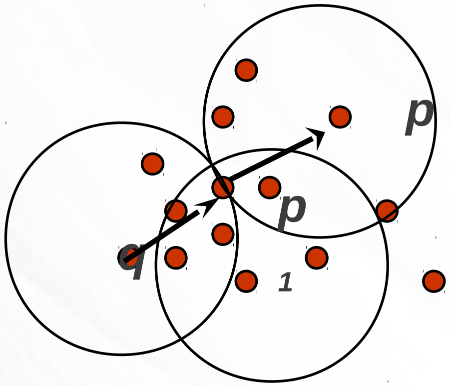
2) core point condition:

|*N_ε (q)*| >= ***MinPts***

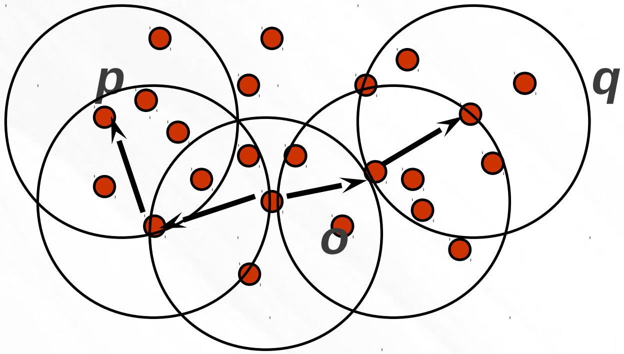
Density-Reachable and Density-Connected

(w.r.t. *Eps*, *MinPts)*

- Let p be a core point, then every point in its *Eps* neighborhood is said to be directly density-reachable from p.



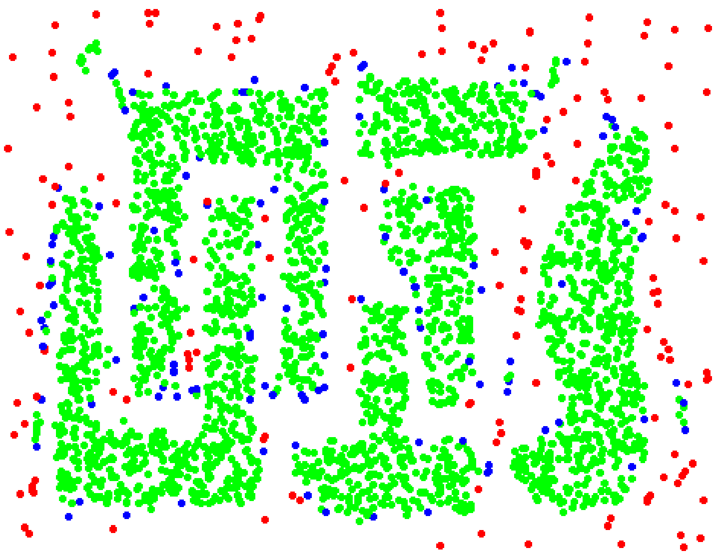
- A point *p* is density-reachable from a point core point *q* if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$



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

DBSCAN: Large Eps

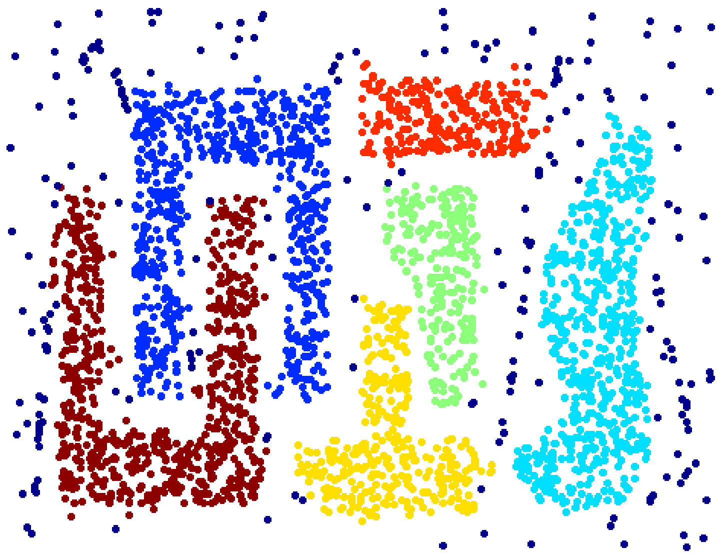
Original Points



Point types: core, border and noise

DBSCAN: Optimal Eps

Original Points

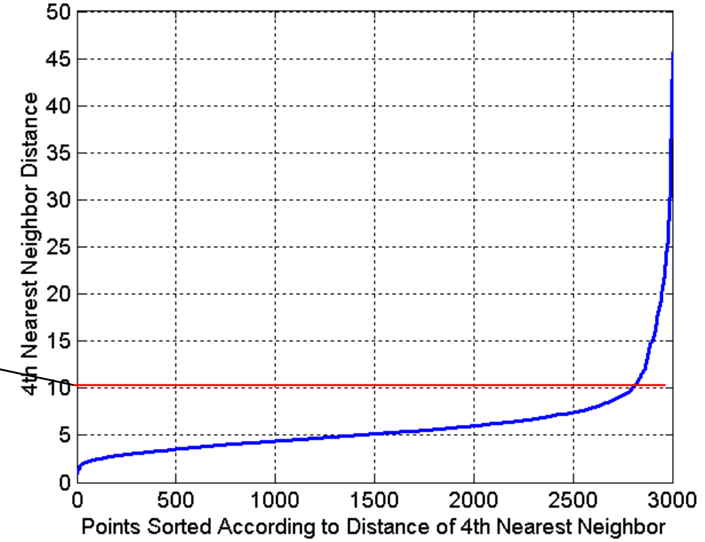


Clusters

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 (e.g., k=4)

Thus, eps=10



DBSCAN: Algorithm

Let ClusterCount=0. For every point *p*:

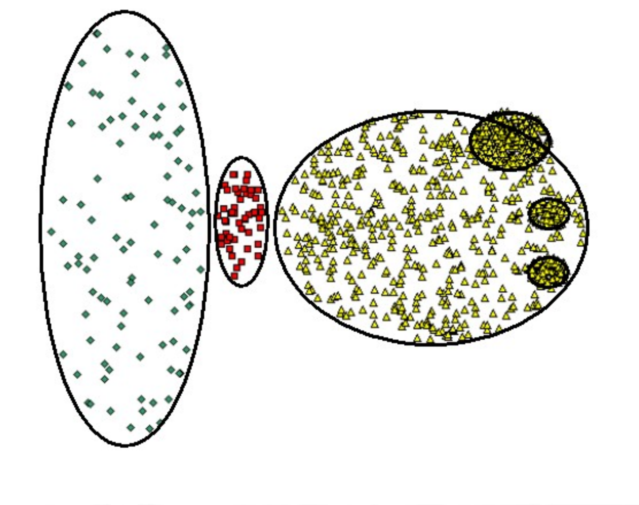
1. If *p* it is not a core point, assign a null label to it [e.g., zero]
2. If *p* is a core point, a new cluster is formed [with label ClusterCount:= ClusterCount+1]

   Then find all points density-reachable from p and classify them in the cluster.
   [Reassign the zero labels but not the others]

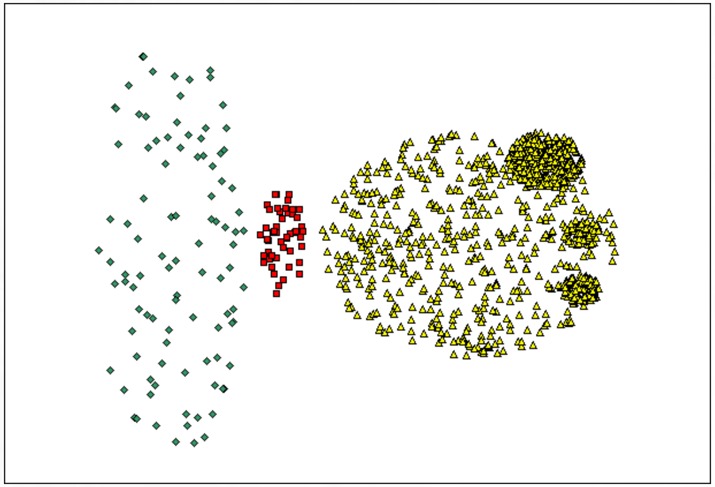
Repeat this process until all of the points have been visited.

Since all the zero labels of border points have been reassigned in 2, the remaining points with zero label are noise.

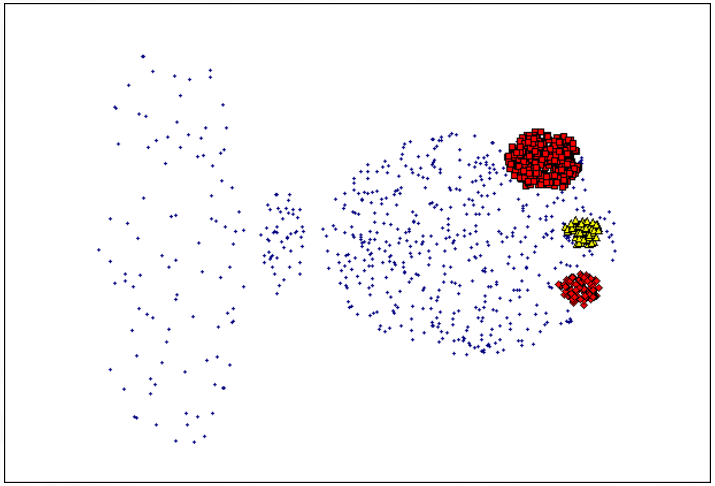
DBSCAN: Flaws



Original Points



(MinPts=4, Eps=large value).



(MinPts=4, Eps=small value; min density increases)

- **Varying densities**
- **High-dimensional data**

DBSCAN: Complexity

Time Complexity: $O(n^2)$—for each point it has to be determined if it is a core point, can be reduced to $O(n*\log(n))$ in lower dimensional spaces by using efficient data structures (n is the number of objects to be clustered);

Space Complexity: $O(n)$.

Resources

- Pang-Ning Tan, Michael Steinbach, Vipin Kumar. Introduction to Data Mining. Michigan State University. University of Minnesota. http://www-users.cs.umn.edu/~kumar/dmbook/index.php
- http://www.cse.ust.hk/~qyang/337/slides/cluster.ppt
- http://www2.cs.uh.edu/~ceick/ML/Topic9.ppt
- www.cs.uiuc.edu/~hanj and Martin Pfeifle www.dbs.informatik.uni-muenchen.de
- http://www.cs.ucla.edu/classes/spring08/cs240B/notes/clusteringCont.ppt

Questions?