Unsupervised Learning
Density-based Methods

Road Map

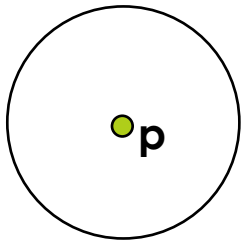
1. DBSCAN
2. OPTICS
3. Subspace Clustering

The Principle

- Regard clusters as dense regions in the data space separated by regions of low density
- Major features
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need of density parameters as termination condition
- Several interesting studies
  - **DBSCAN**: Ester, et al. (KDD'96)
  - **OPTICS**: Ankerst, et al (SIGMOD'99).
  - **DENCLUE**: Hinneburg & D. Keim (KDD'98)
  - **CLIQUE**: Agrawal, et al. (SIGMOD'98) (more grid-based)

ε-neighborhood & core objects

- The neighborhood within a radius ε of a given object is called the ε-**neighborhood** of the object



ε = 1 cm

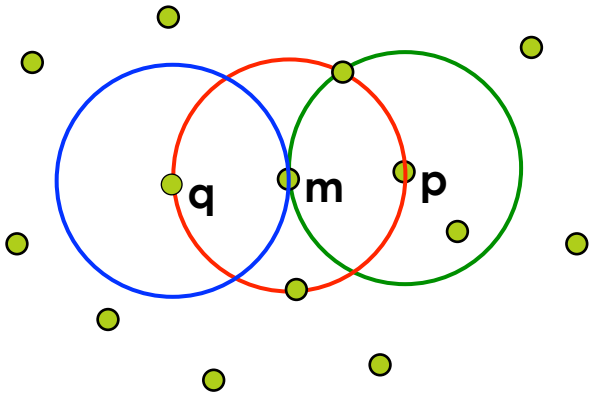
- If the ε-neighborhood of an object contains at least a minimum number, **MinPts**, of objects than the object is called a **core object**

  - **Example:** ε = 1 cm, MinPts=3

m and **p** are core objcets because

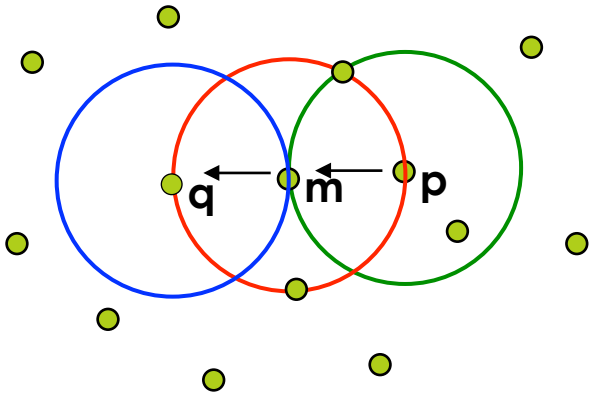
Each is in an ε-neighborhood

containing at least 3 points



Directly Density-Reachable Objects

- An object **p** is **directly density-reachable** from object **q** if p is within the the ε-neighborhood of q and *q* is a core object



- **Example:**

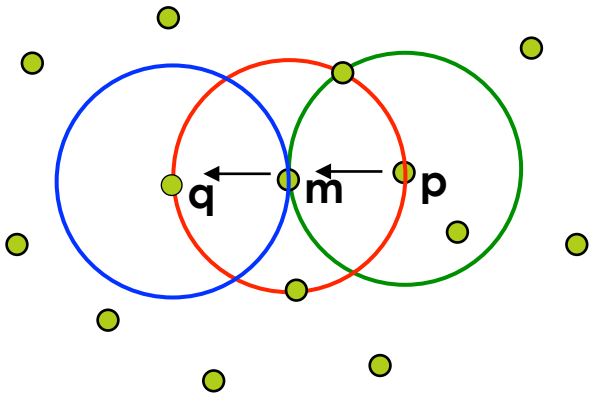
q is directly density-reachable from **m**

m is directly density-reachable from **p**

and vice versa

Density-Reachable Objects

- An object **p** is **density-reachable** from object **q** with respect to **ε** and **MinPts** if there is a chain of objects $p_1,\ldots p_n$ where $p_1=q$ and $p_n=p$ such that p_{i+1} is directly reachable from p_i with respect to ε and MinPts



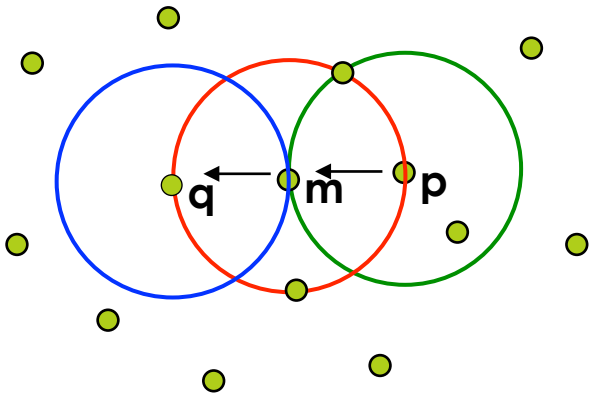
- **Example:**

  q is density-reachable from **p** because **q** is directly density-reachable from **m** and **m** is directly density-reachable from **p**

  p is not density-reachable from **q** because **q** is not a core object

Density-Connectivity

- An object **p** is **density-connected** to object **q** with respect to **ε** and **MinPts** if there is an object **O** such as both p and q are density reachable from **O** with respect to ε and MinPts



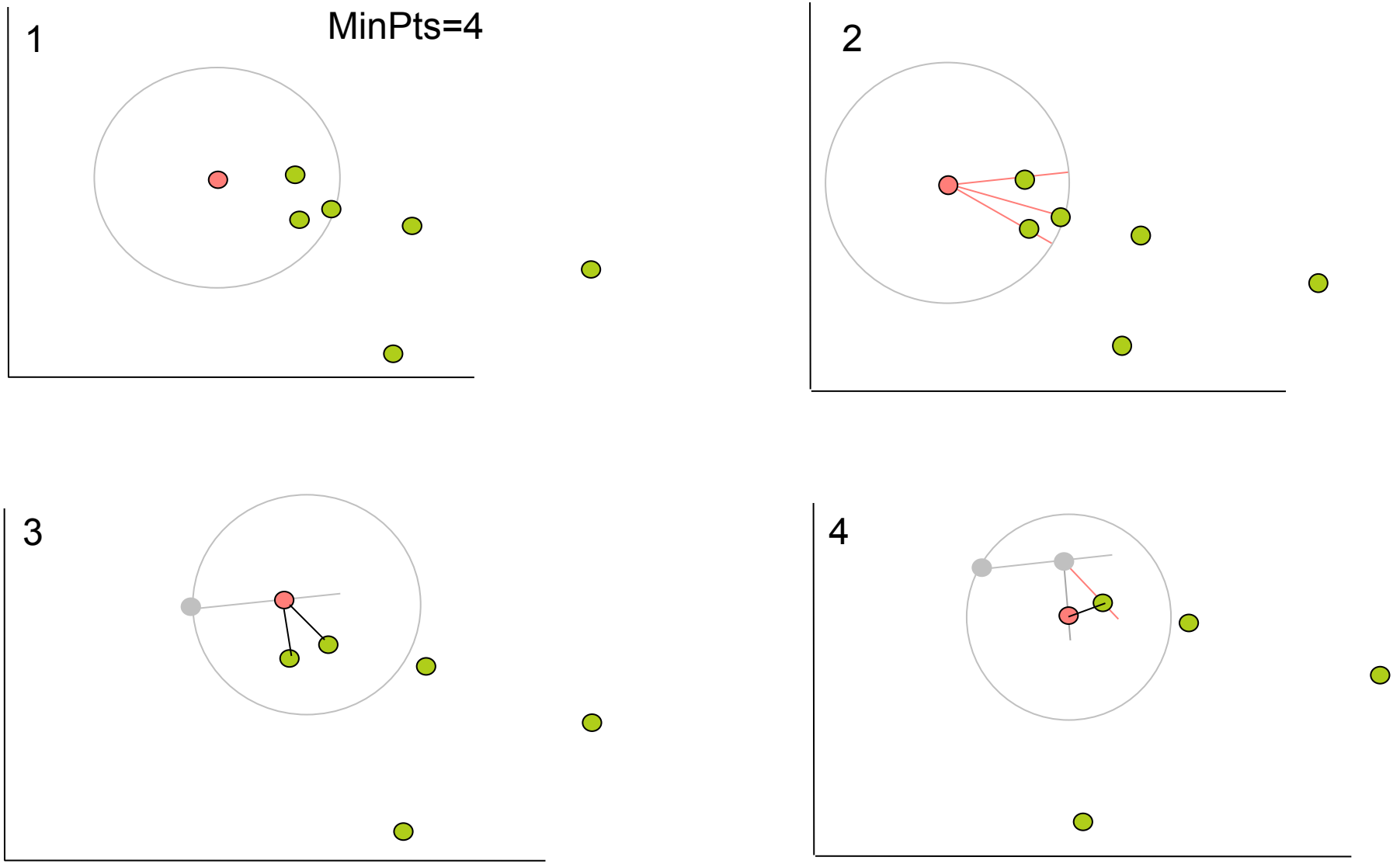
- **Example:**

  p,q and **m** are all density connected

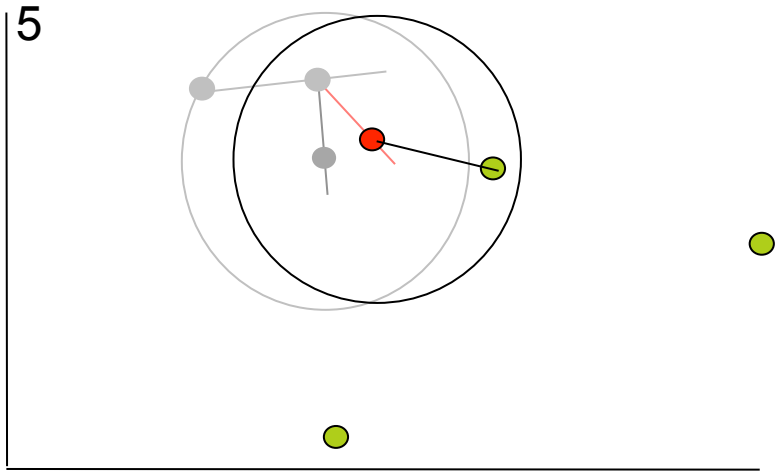
DBSCAN Algorithm

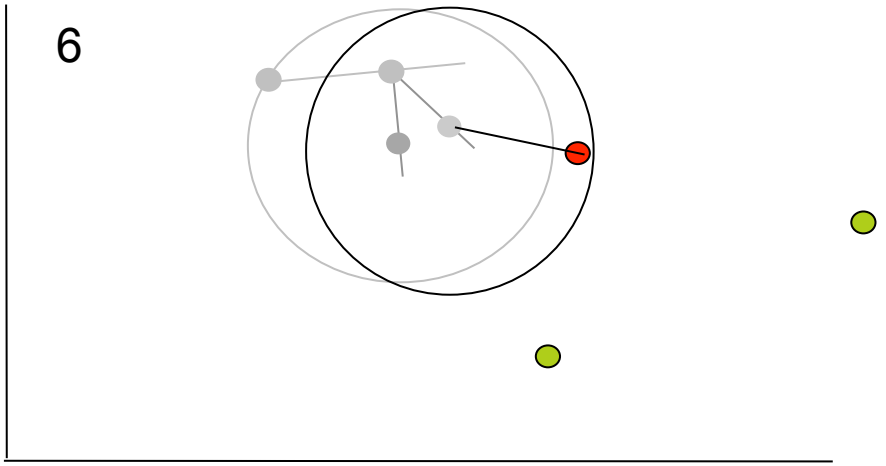
- Searches for clusters by checking the ε-neighborhood of each point in the database
- If the ε-neighborhood of a point p contains more than MinPts, a new cluster with p as a core object is created
- DBSCAN iteratively collects directly density reachable objects from these core objects. Which may involve the merge of a few density-reachable clusters
- The process terminates when no new point can be added to any cluster

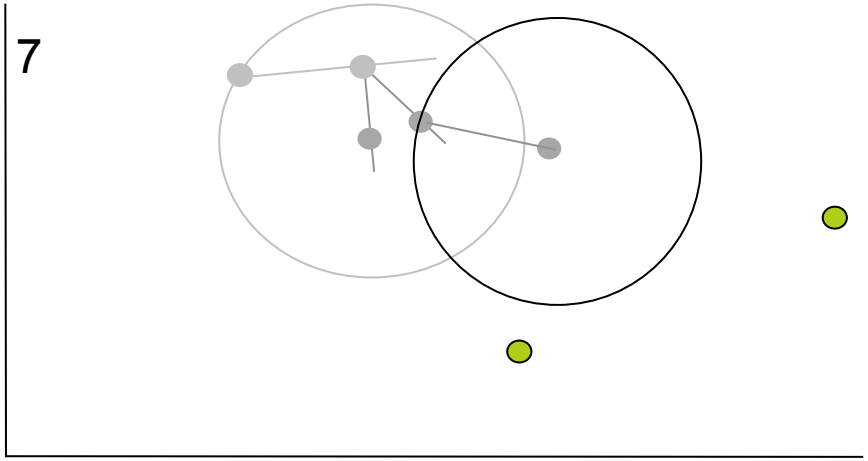
DBSCAN Example

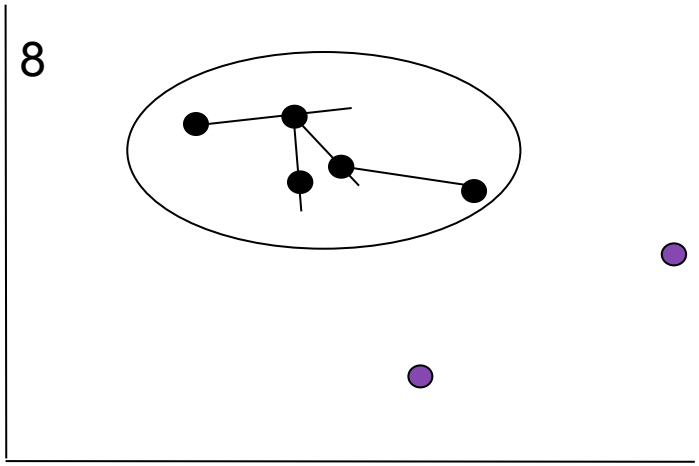


DBSCAN Example

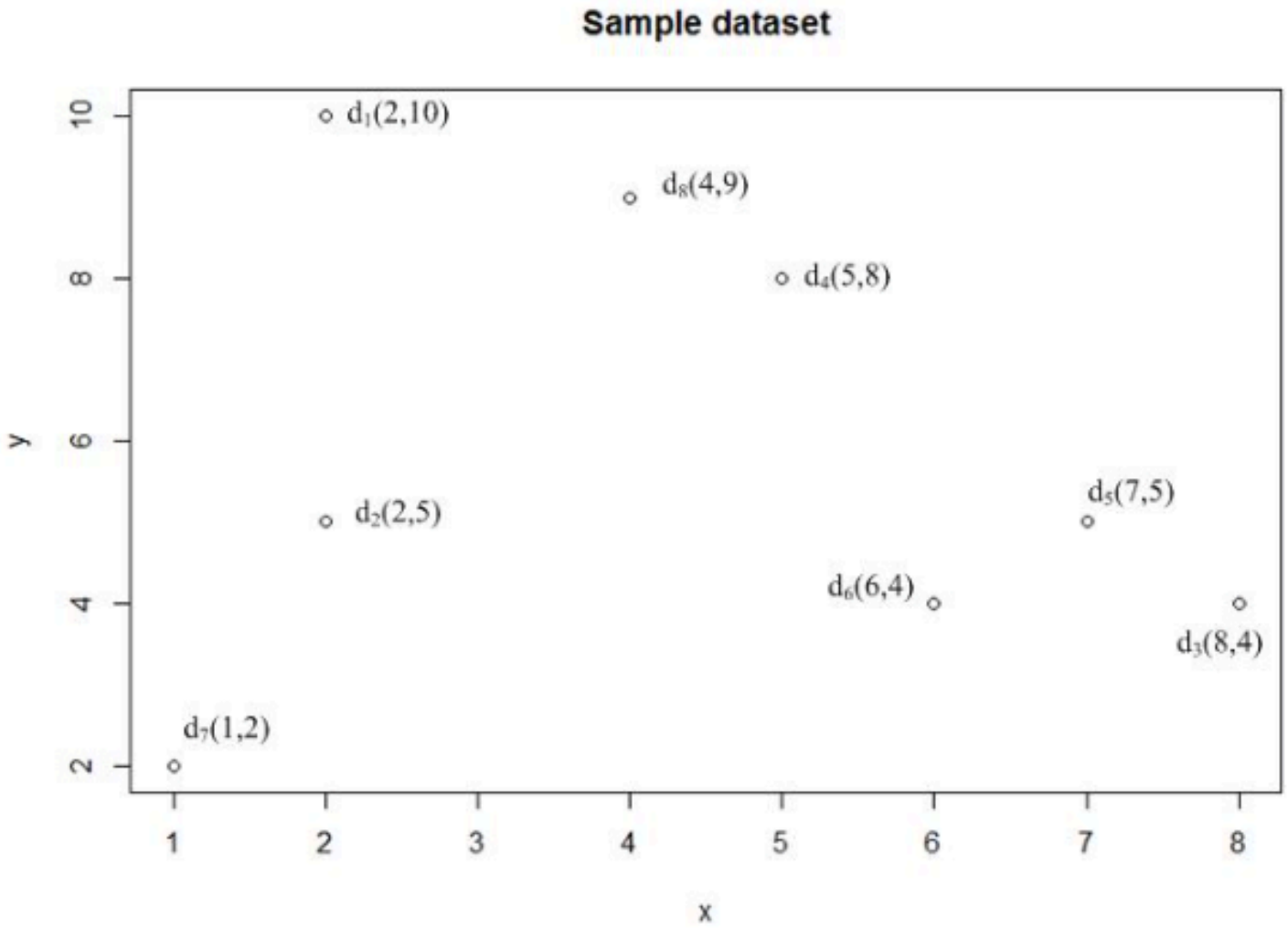








Exercise

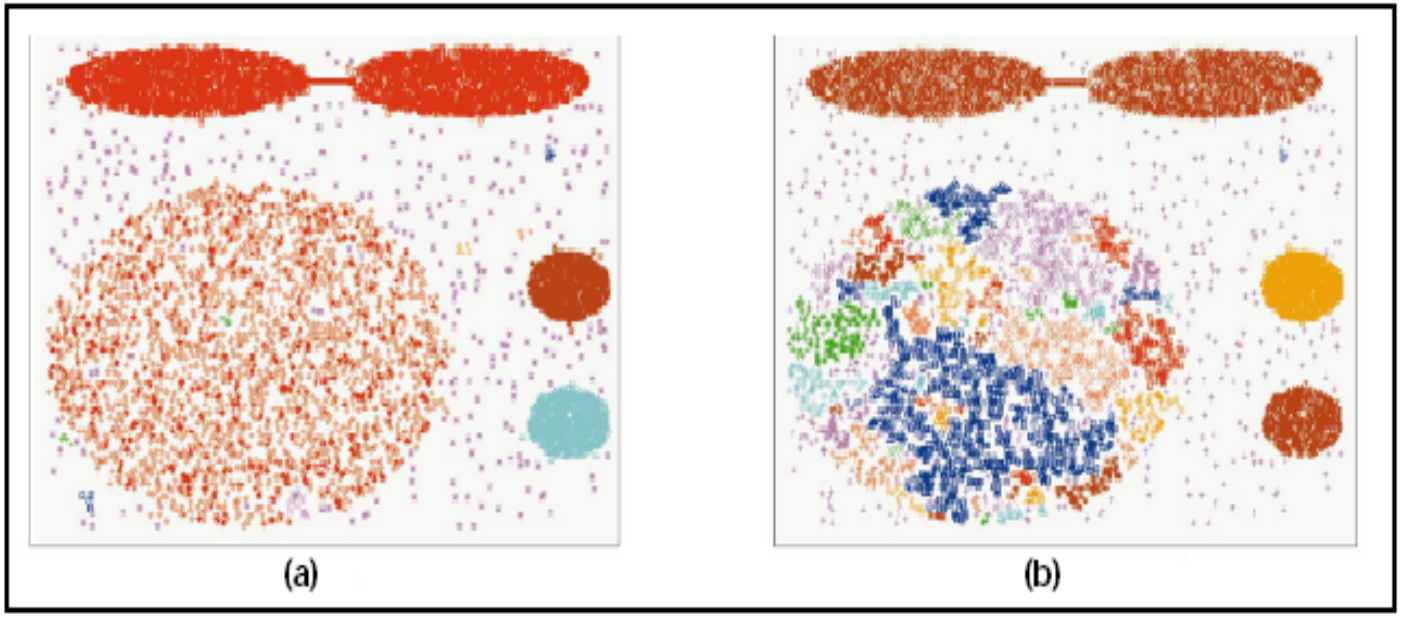

Sample dataset
d1(2,10)
d8(4,9)
d4(5,8)
d2(2,5)
d5(7,5)
d6(6,4)
d3(8,4)
d7(1,2)
y
x
2
4
6
8
10
1
2
3
4
5
6
7
8

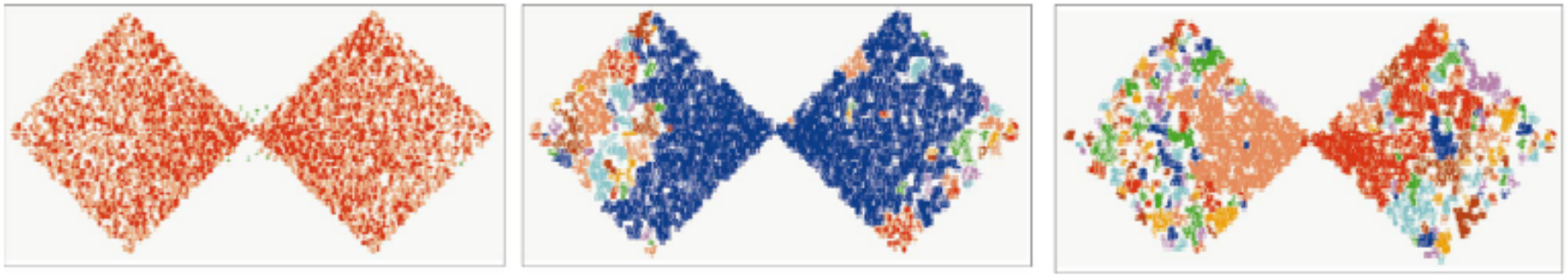
Distance

$dist(d_i, d_j)$	1	2	3	4	5	6	7	8
1	0	5	8.5	3.6	7.1	7.2	8.1	2.2
2		0	6.1	4.2	5	4.1	3.2	4.5
3			0	5	1.4	2	7.3	6.4
4				0	3.6	4.1	7.2	1.4
5					0	1.4	6.7	5
6						0	5.4	5.4
7							0	7.6
8								0

DBSCAN: Sensitive to Parameters

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



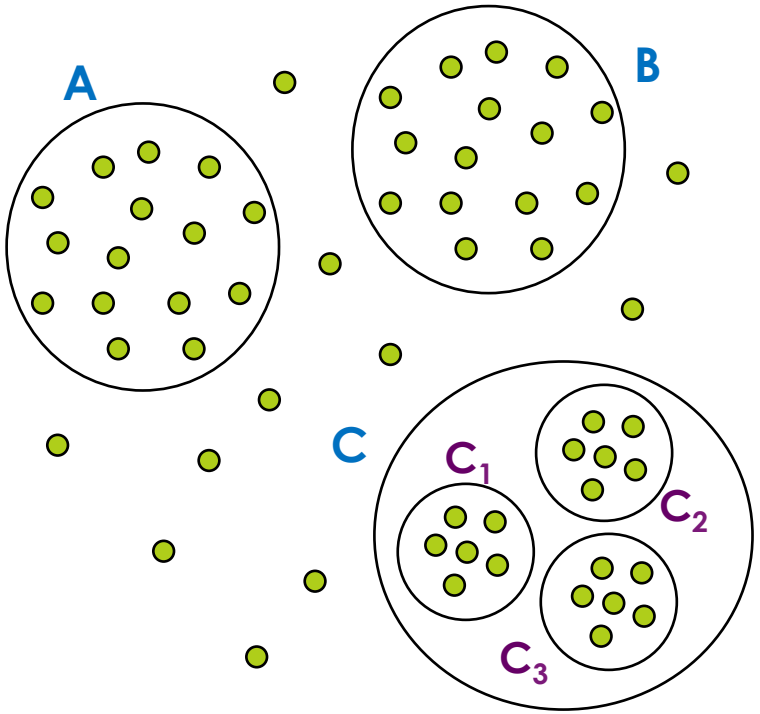


Road Map

1. DBSCAN
2. OPTICS
3. Subspace Clustering

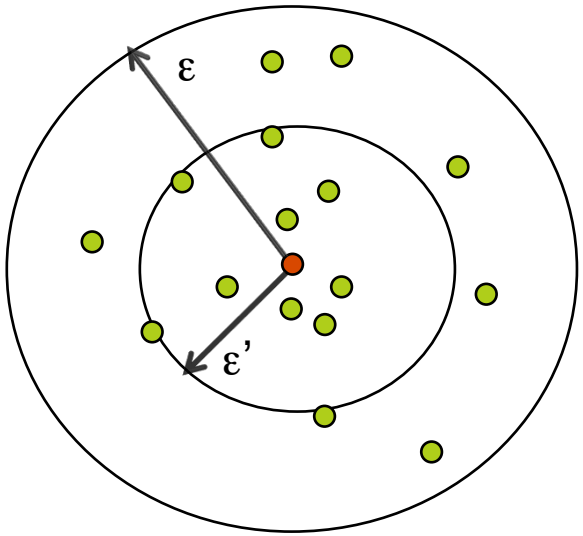
Why DBSCAN is not enough?

- Very different local densities may be needed to reveal clusters in different regions
- Clusters **A**,**B**,C_1,C_2, and C_3 cannot be detected using one global density parameter
- A global density parameter can detect either **A**,**B**,**C** or C_1,C_2,C_3
- **Solutions**
  - Use hierarchical clustering, but
    - Single link effect
    - Hard to interpret
  - Use OPTICS



OPTICS Principle

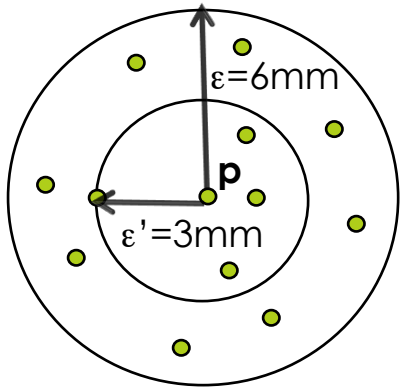
- Produce a special order of the database
  - with respect to its density-based clustering structure
  - containing information about every clustering level of the data set (up to a generating distance ε)



  - Which information to use?

Core-distance and Reachability-distance

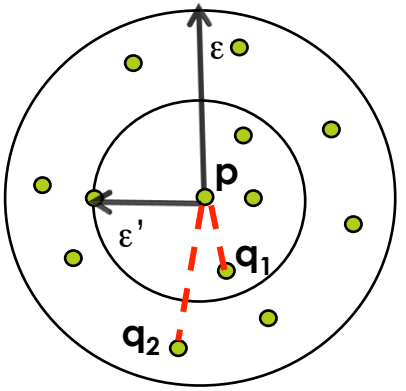
- The **core-distance** of an object **p** is the smallest **ε'**that makes **p** a core object
  - If **p** is not a core object, the core distance of **p** is **undefined**
  - **Example** (ε, MinPts=5)
    - ε' is the core distance of p
    - It is the distance between p and the fourth closest object



- The **reachability-distance** of an object **q** with respect to object to object **p** is:

Max(core-distance(p), Euclidian(p,q))

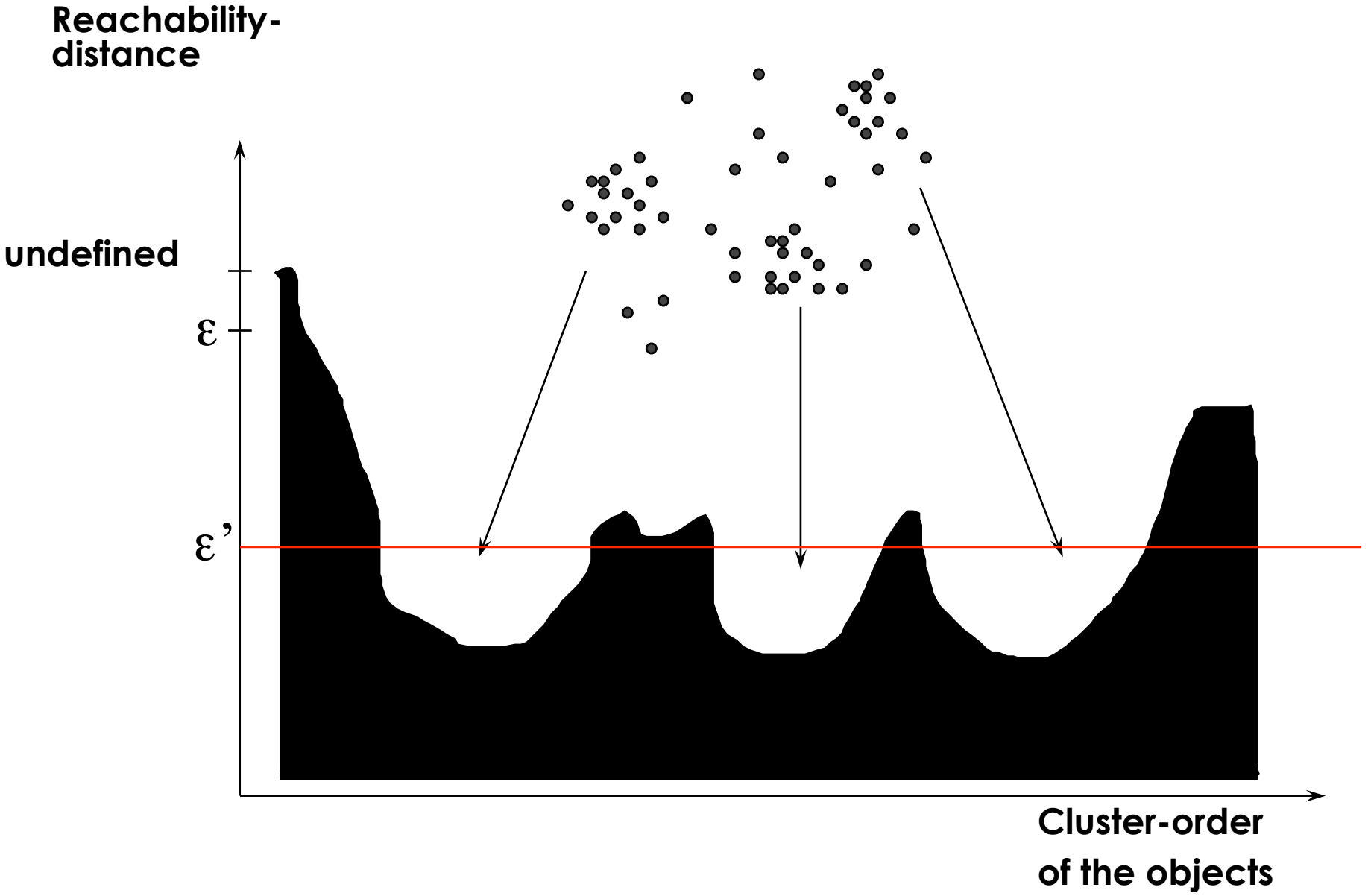
  - **Example**
    - Reachability-distance(q_1,p)=core-distance(p)=ε'
    - Reachability-distance(q_2,p)=Euclidian(q_2,p)



OPTICS Algorithm

- Creates an ordering of the objects in the database and stores for each object its:
  - Core-distance
  - Distance reachability from the closest core object from which an object have been directly density-reachable
- This information is sufficient for the extraction of all density-based clustering with respect to any distance ε' that is smaller than ε used in generating the order

Illustration of Cluster Ordering

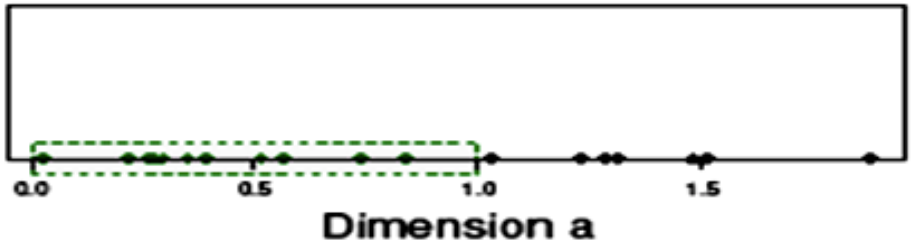


Road Map

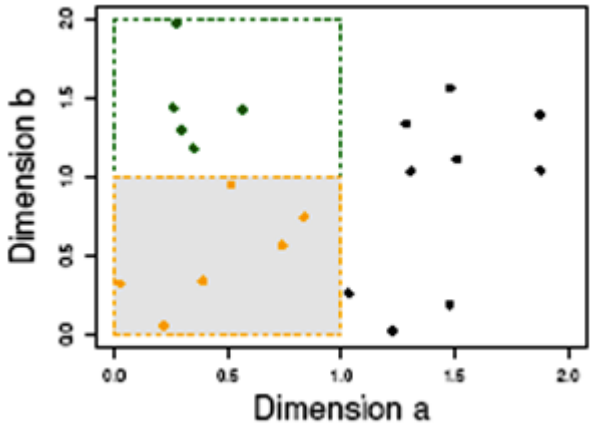
1. DBSCAN
2. OPTICS
3. Subspace Clustering

The Curse of Dimensionality

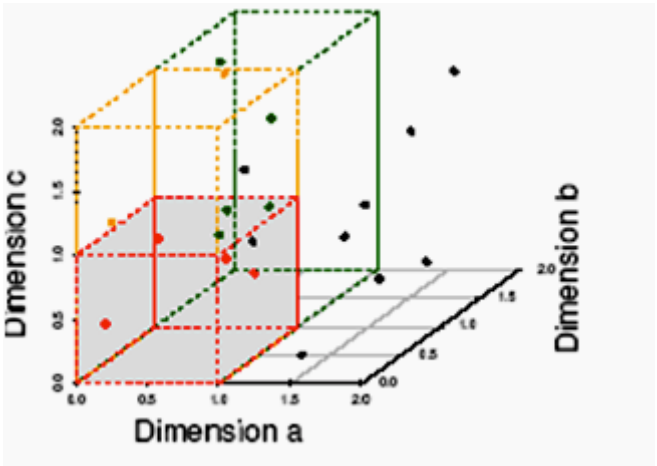
Let's take an example of one dimensional data



We add a second dimension

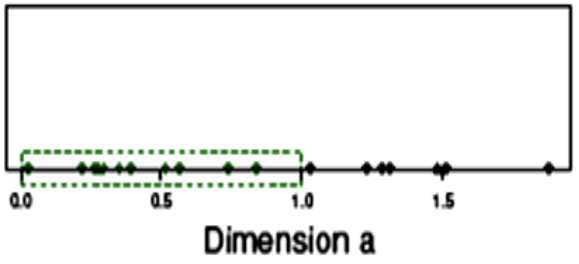


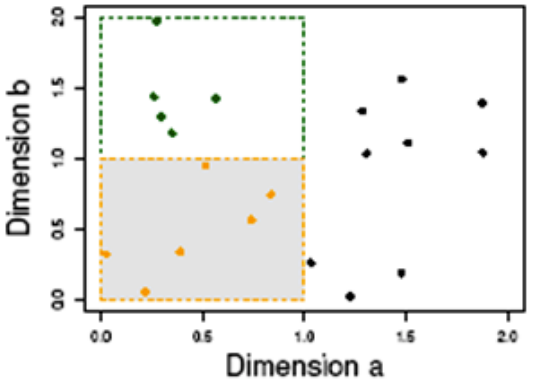
We add a third dimension

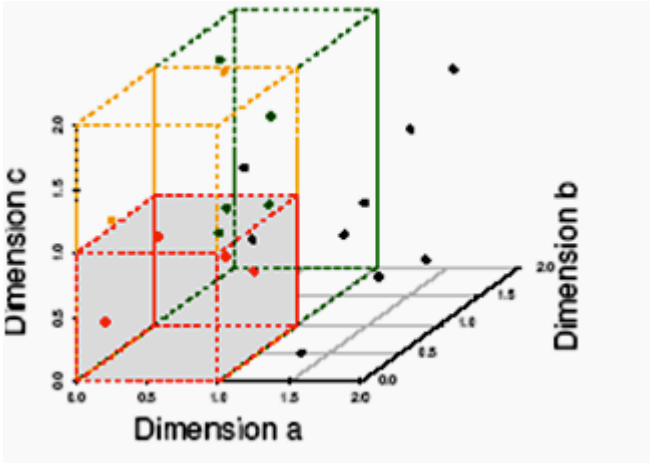


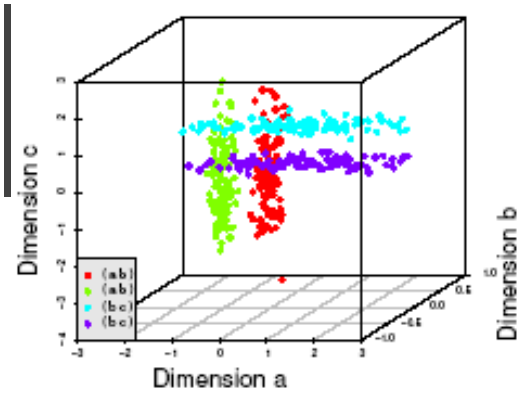
The Curse of Dimensionality

- Data in only one dimension is relatively packed
- Adding a dimension "stretch" the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless (equidistance)



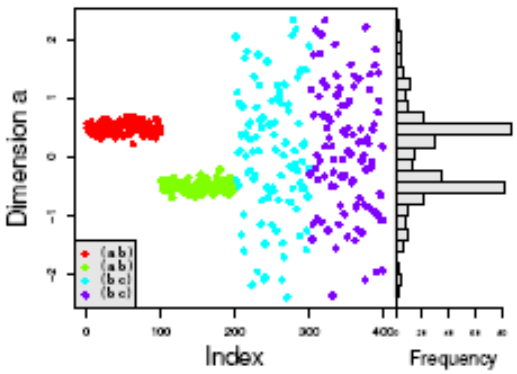




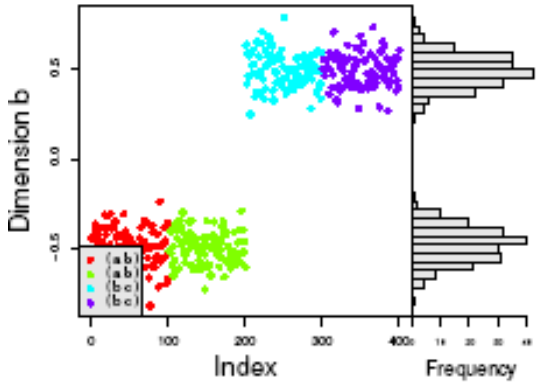


Subspace Clustering

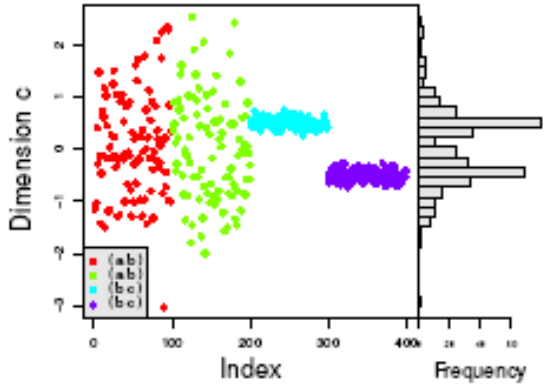
- Extension to attribute selection
- Clusters may exist only in some subspaces
- **Subspace-clustering:** find clusters in all the subspaces



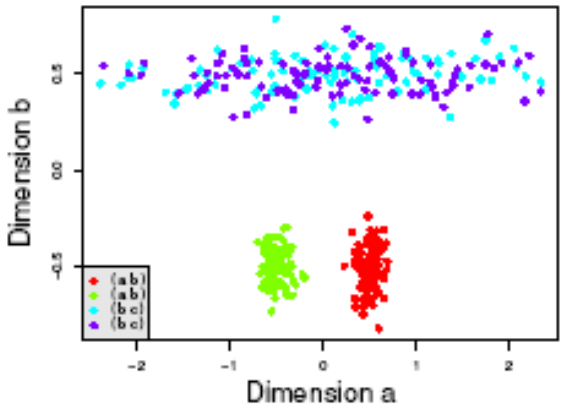
(a) Dimension a



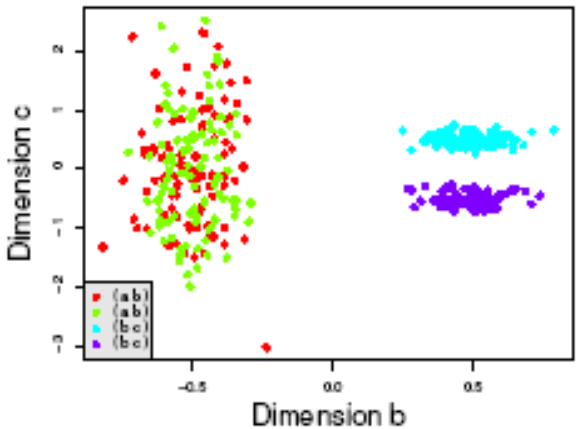
(b) Dimension b



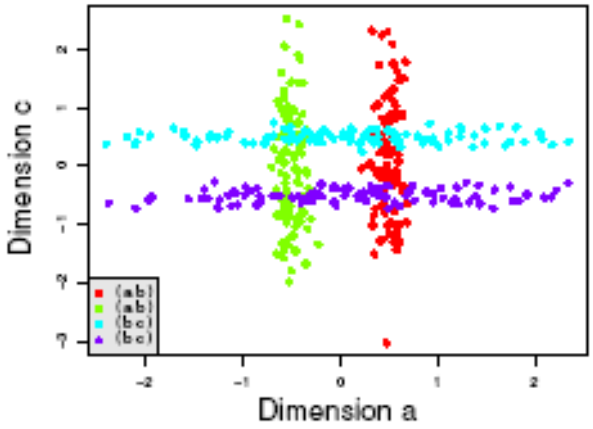
(c) Dimension c



(a) Dims a & b



(b) Dims b & c



(c) Dims a & c

Subspace Clustering

- How to find subspace clusters effectively and efficiently?

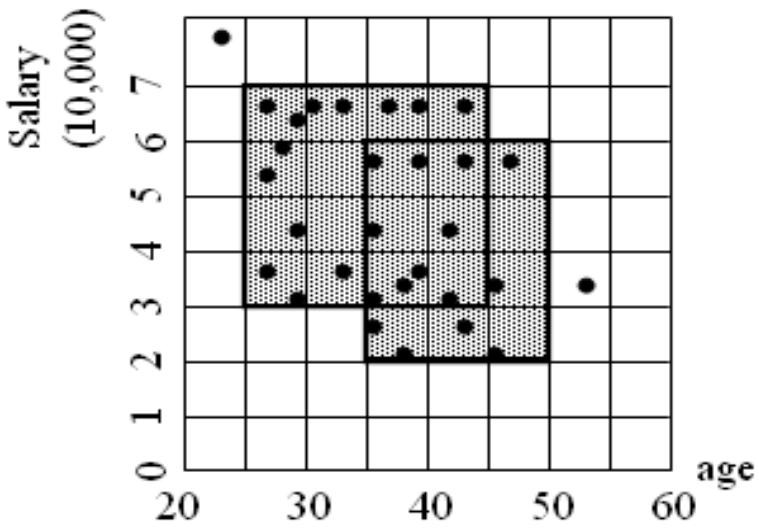
- We are going to see two approaches
  - Dimension-growth subspace clustering
  - Frequent pattern-based clustering

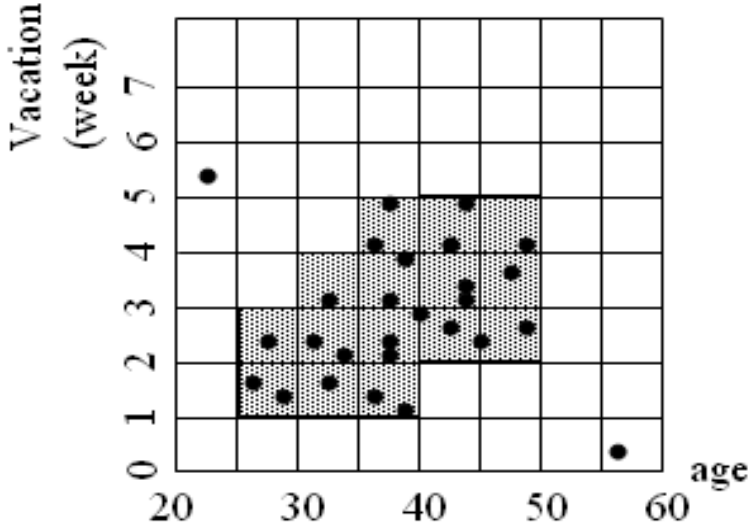
CLIQUE

- **CLIQUE (CLustering in QUEst)** was the first algorithm proposed for dimension **growth subspace clustering** in high-dimensional space
- Start at single-dimensional subspaces and grow upward to higher dimensional ones
- CLIQUE partitions each dimension like a grid structure and determines whether a cell is dense based on the number of points it contains
- CLIQUE is an integration of grid-based and density-based methods

CLIQUE

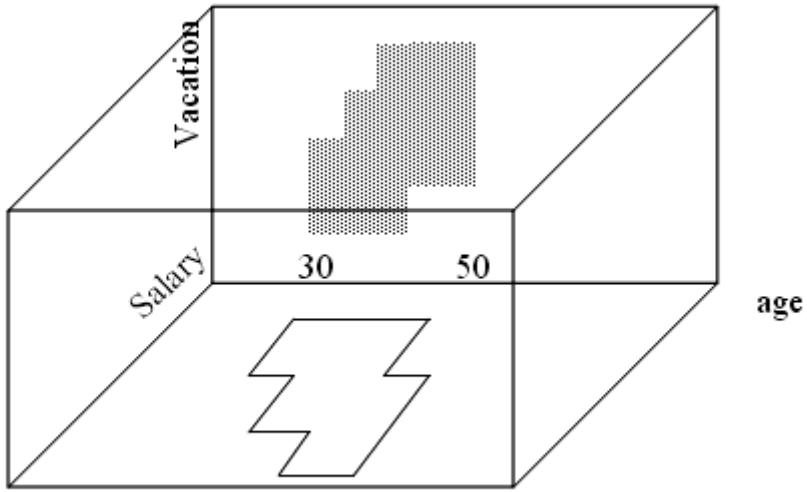
- Partition the d-dimensional data space into non overlapping rectangular units (done in 1-D for each partition)
- Identify dense units
- A unit is dense if the fraction of total data points contained in it exceeds an input model parameter





CLIQUE

- The subspaces representing dense regions are intersected to form a **candidate** search space in which dense units of higher dimensionality may exist



- Why does CLIQUE confine its search for dense units of higher dimensionality to the intersection of the dense units in the subspaces?

CLIQUE

- The property adapted by CLIQUE states:
  - If a k-dimensional unit is dense, then so are its projections in (k-1) dimensional space
- Generate potential or candidate dense units in k-dimensional space from dense units found in (k-1) dimensional space
- The resulting space searched is much smaller than the original space
- The dense units are then examined to determine clusters

CLIQUE

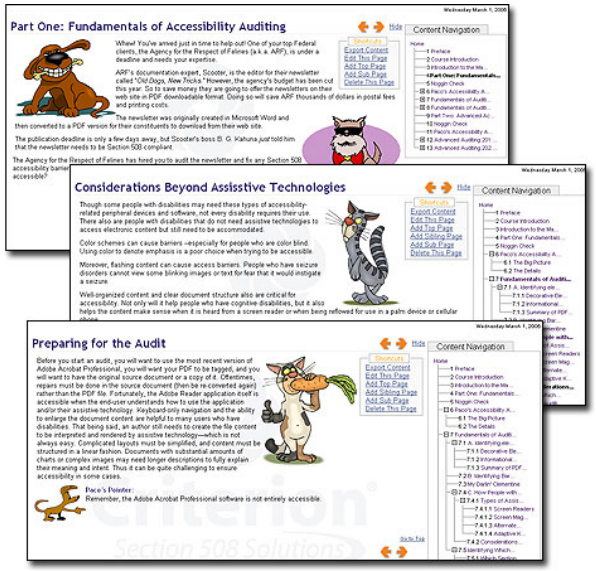
- **Strength**
  - Automatically finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
  - Insensitive to the order of records in input and does not presume some canonical data distribution
  - Scales linearly with the size of input and has good scalability as the number of dimensions in the data increases
- **Weakness**
  - The accuracy of the clustering result may be degraded at the expense of simplicity of the method

Frequent Pattern-based Clustering

- Frequent pattern mining leads to the discovery of interesting associations and correlations among data objects
- The frequent patterns discovered may also indicate clusters
- Well suited for high dimensional data
  - Rather than growing clusters dimension by dimension, we grow sets of frequent items
  - Lead to clusters descriptions

Example: Frequent term-based text

- Documents contain terms
- Extract terms
  - Parsing
  - Stemming
- Each document can be represented as a set of terms
- Consider each term as a dimension
- The dimension space will be very high
- The dimension space can be referred as: **term vector space**



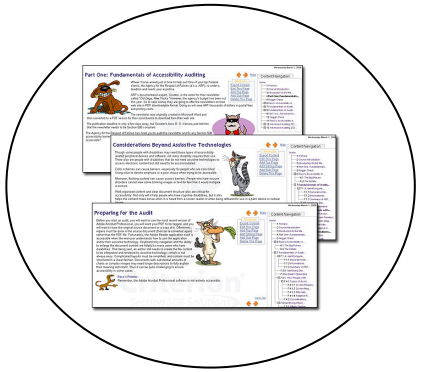
Example: Frequent term-based text

- Documents are clustered based on the frequent terms they contain
- Consider only the low-dimensional frequent term sets as "cluster candidates"
- Frequent term set is not a cluster but a description of a cluster
- A cluster consists of documents containing all the terms of the frequent term set

News, education , sport

Cluster1

Science, Computer



Cluster2

Example: Frequent term-based text

- How to select a good subset of the set of all frequent term sets?
- Let
  - $\mathbf{F_i}$ be a set of frequent term sets
  - **Cov($\mathbf{F_i}$)** be the set of documents covered by $\mathbf{F_i}$
- Find a well-selected subset $F_1, F_2, \ldots, F_k$, of all frequent term sets
- **Principle**
  - (1) the selected subset should cover all the documents to be clustered

$$\sum_{i=1}^{k} \mathrm{cov}(F_i) = D$$

  - (2) the overlap between any two partitions Fi and Fj for ($i \neq j$) should be minimized (e.g., using entropy)
- This approach automatically generates cluster description, In traditional methods, an additional step is required to describe the resulting clusters.

Summary

- Density-based Clustering find clusters with arbitrary shapes
- Handles noise
- Handles High Dimensional Data