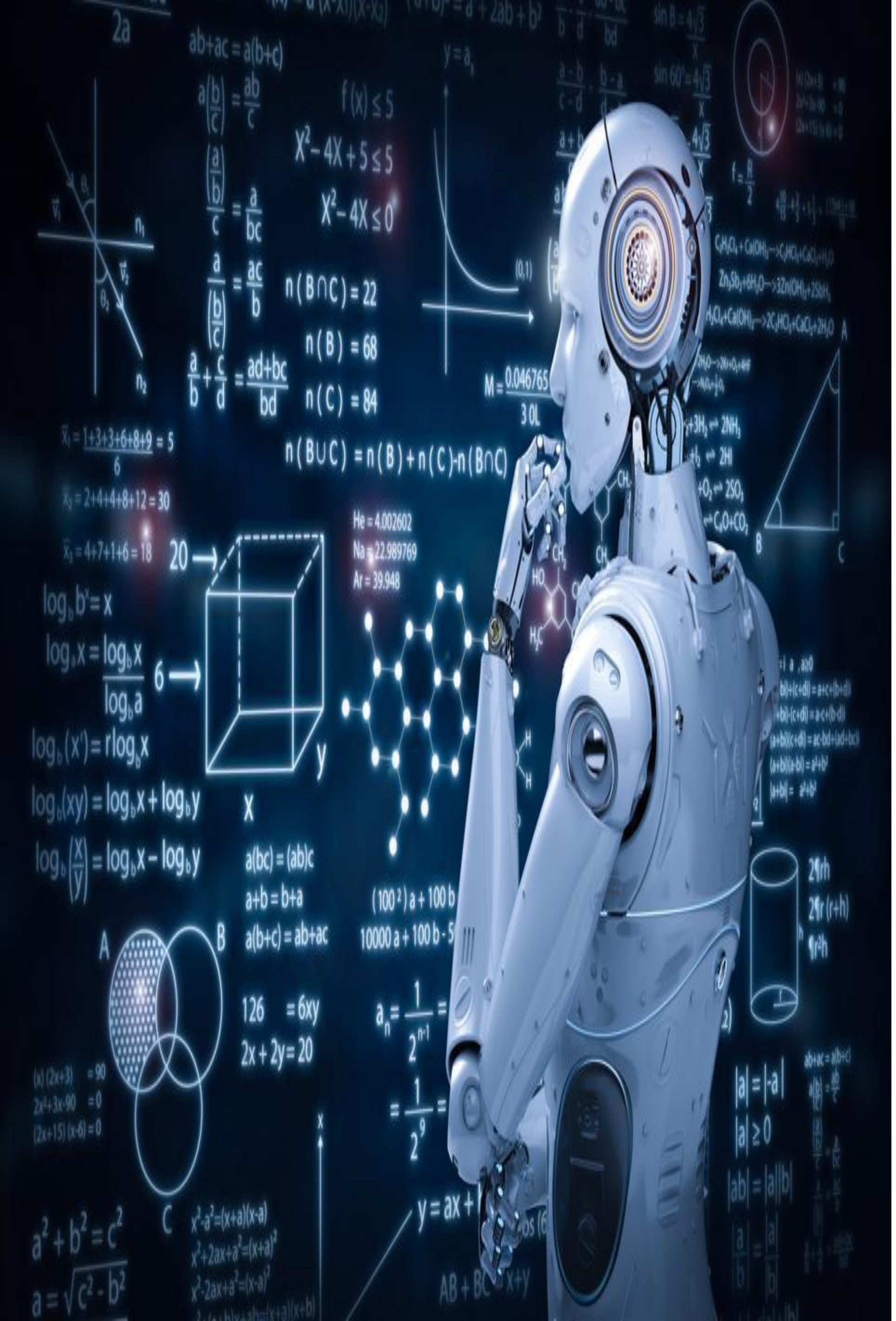


Practical Machine Learning

Day 11: Mar24 DBDA

Kiran Waghmare

Agenda

- Clustering
- K-Means
- https://www.naftaliharris.com/blog/visualizing-k-means-clustering/
- Hierarchical
- DB-SCAN

Concepts: Preliminary

- **DBSCAN is a density-based algorithm**

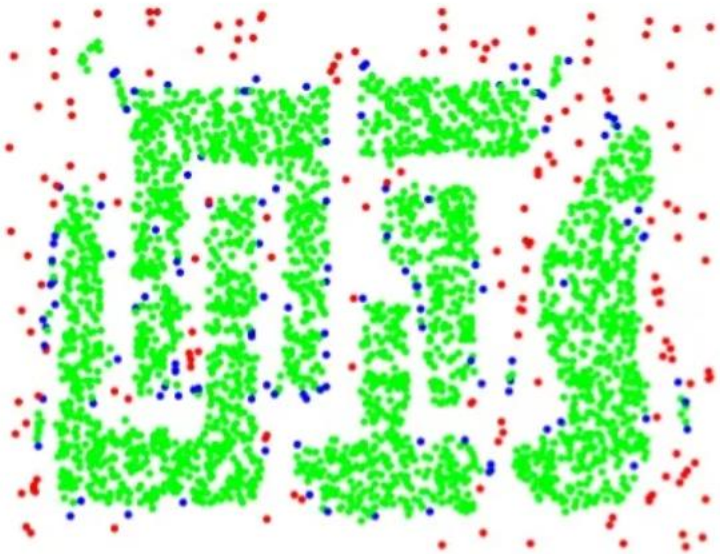
- DBScan stands for Density-Based Spatial Clustering of Applications with Noise

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

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

Concepts: Preliminary

Original Points



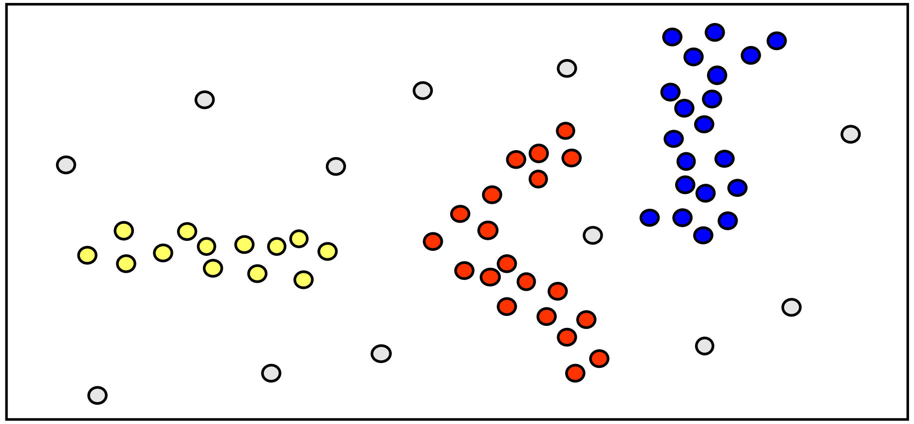
Point types: core, border and noise

Eps = 10, MinPts = 4

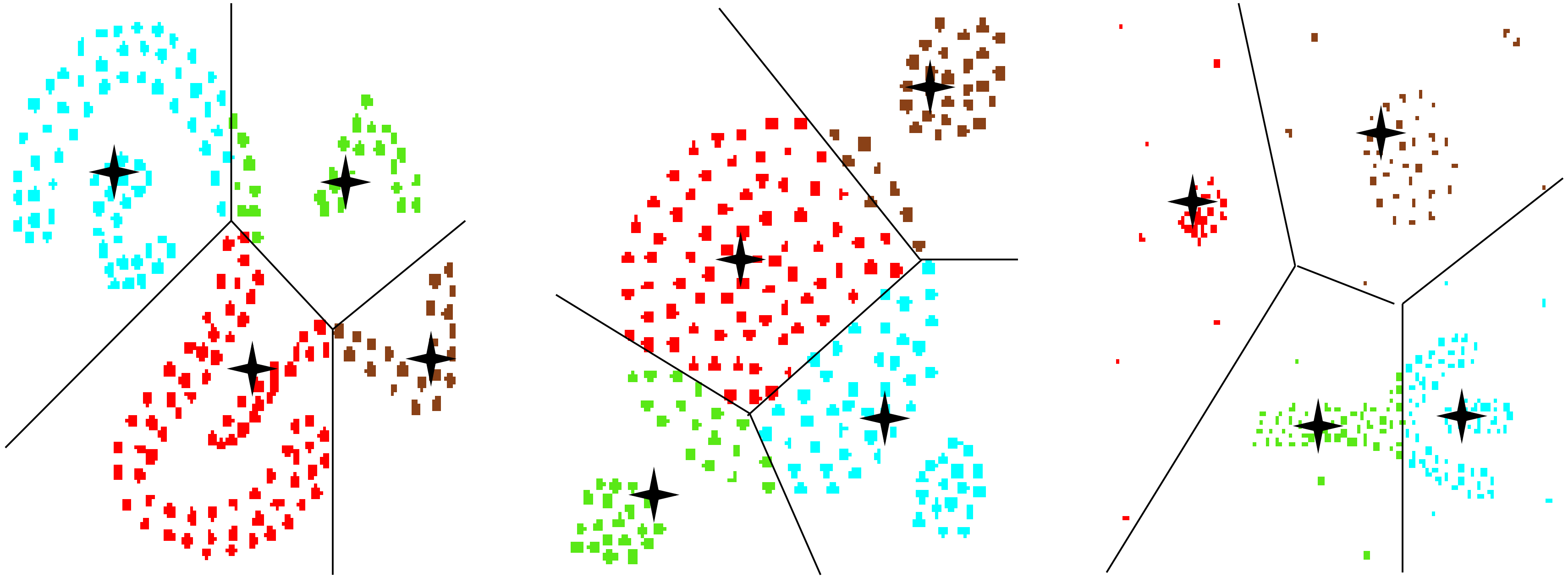
Density-Based Clustering

- *Basic Idea*:

  Clusters are dense regions in the data space, separated by regions of lower object density



- **Why Density-Based Clustering?**



Results of a *k*-medoid algorithm for *k*=4

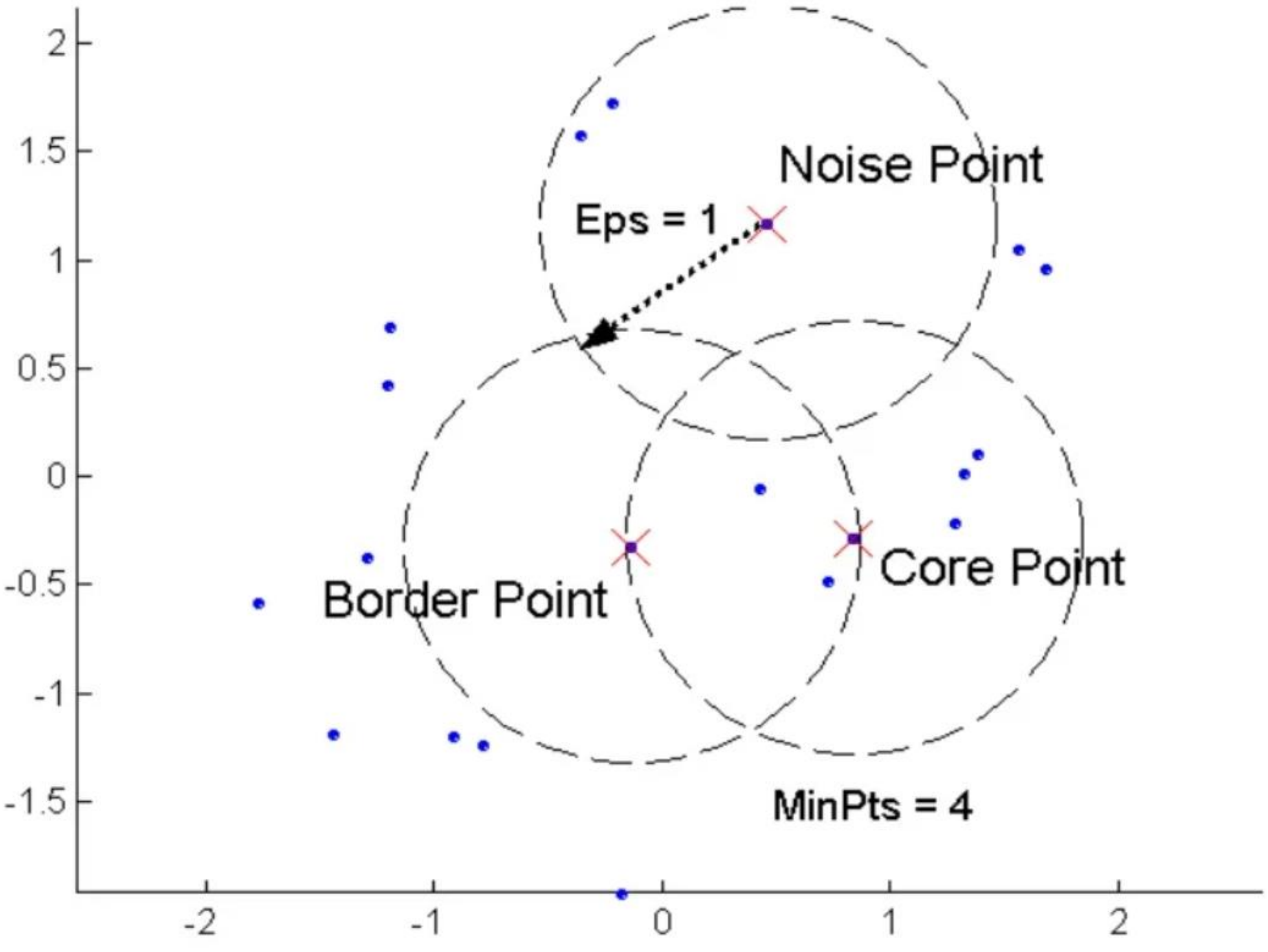
Concepts: Preliminary

- 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

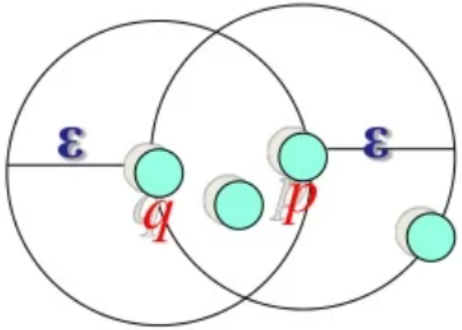
- A noise point is any point that is not a core point or a border point

Concepts: Core, Border, Noise



Concepts: ε-Neighborhood

- ε-Neighborhood - Objects within a radius of ε from an object. (epsilon-neighborhood)
- Core objects - ε-Neighborhood of an object contains at least MinPts of objects



ε-Neighborhood of *p*

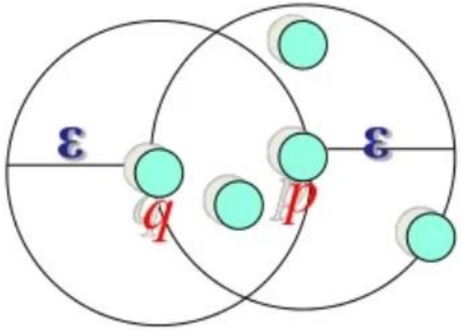
ε-Neighborhood of *q*

p is a core object (MinPts = 4)

q is not a core object

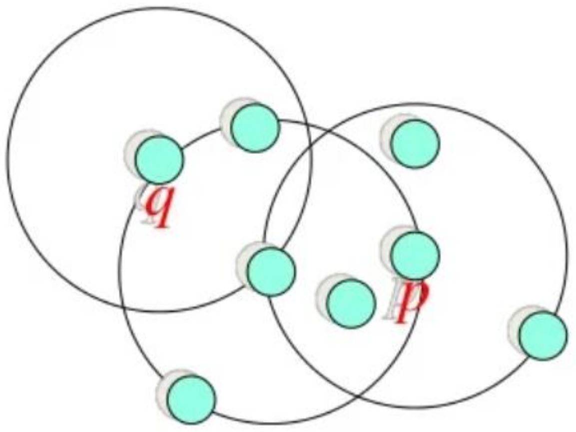
DBScan : Reachability

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

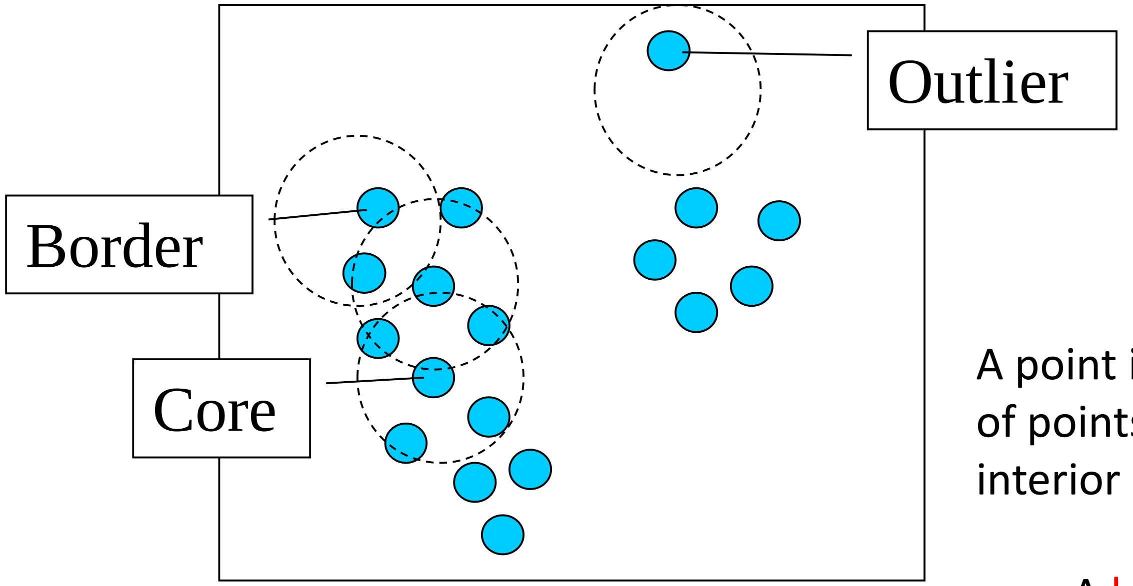


- q is directly density-reachable from p
- p is not directly density-reachable from q.

DBScan : Reachability



Core, Border & Outlier



ε = 1unit, MinPts = 5

Given ε and *MinPts*, categorize the objects into three exclusive groups.

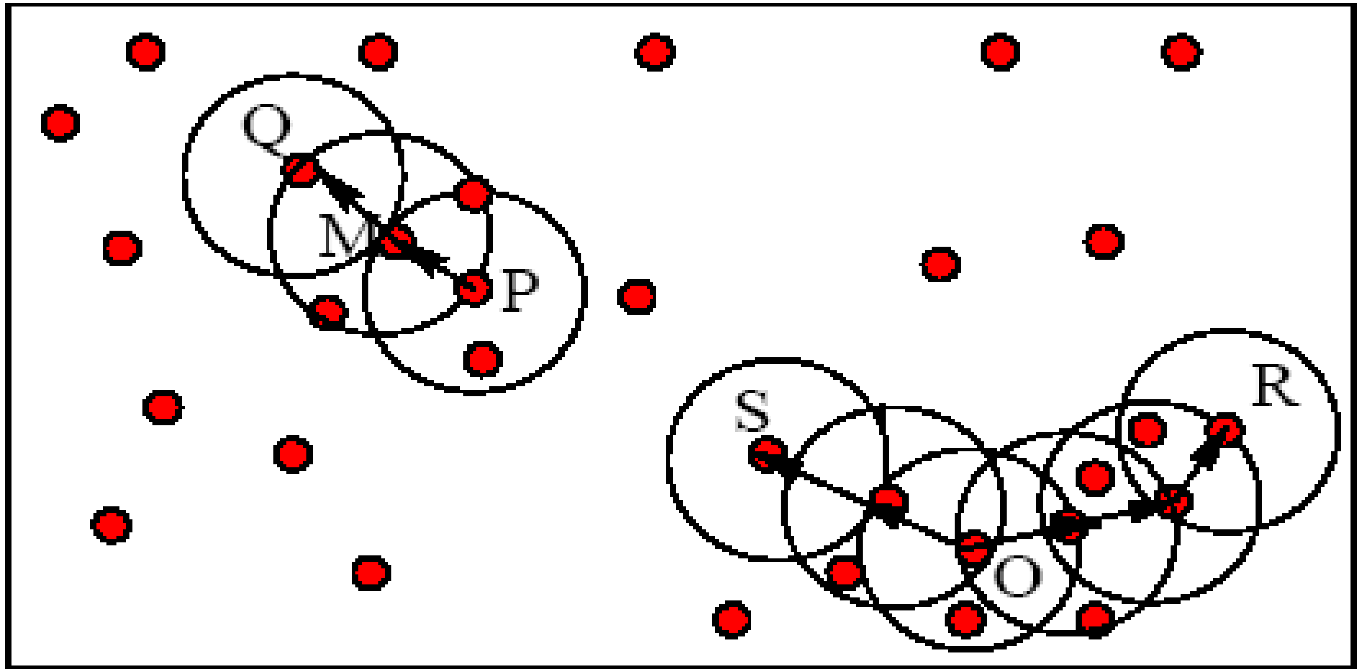
A point is a core point if it has more than a specified number of points (MinPts) within Eps These are points that are at the interior of a cluster.

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 nor a border point.

Example

- M, P, O, and R are core objects since each is in an Eps neighborhood containing at least 3 points

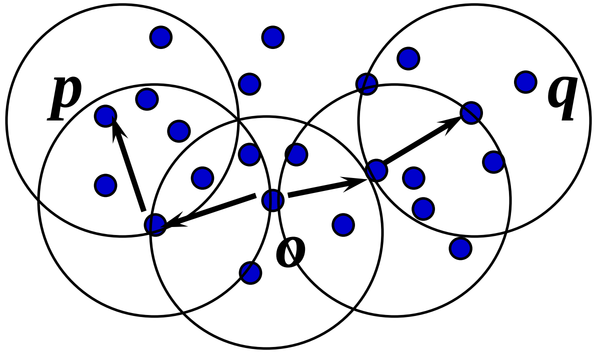


Minpts = 3

Eps=radius of the circles

Density-Connectivity

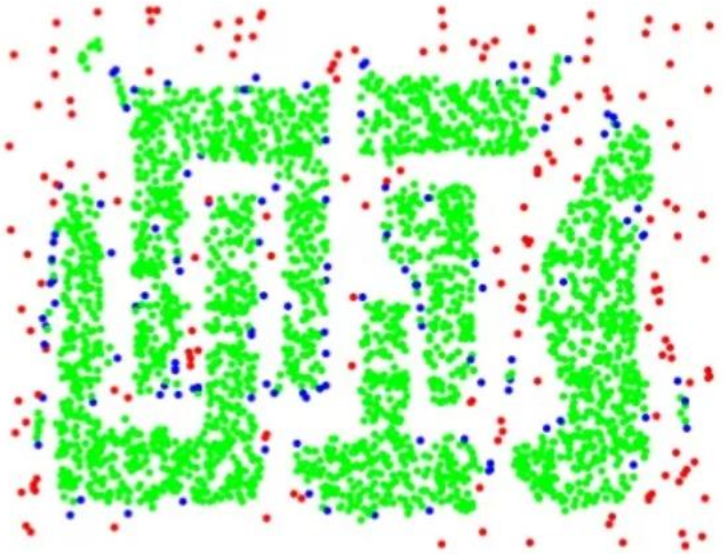
- **Density-reachable is not symmetric**
  - **not good enough to describe clusters**
- **Density-Connected**
  - **A pair of points p and q are density-connected if they are commonly density-reachable from a point o.**



- **Density-connectivity is symmetric**

Core, Border, Noise points representation

Original Points

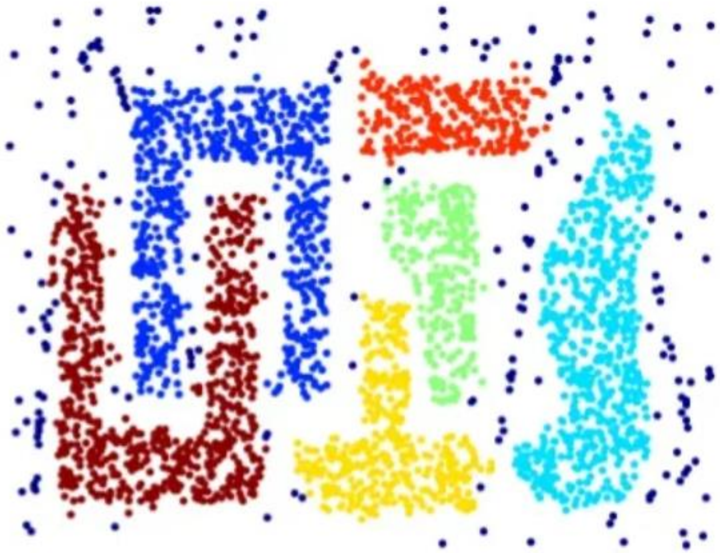


Point types: core, border and noise

Eps = 10, MinPts = 4

Clustering

Original Points



Clusters

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

DBScan Algorithm

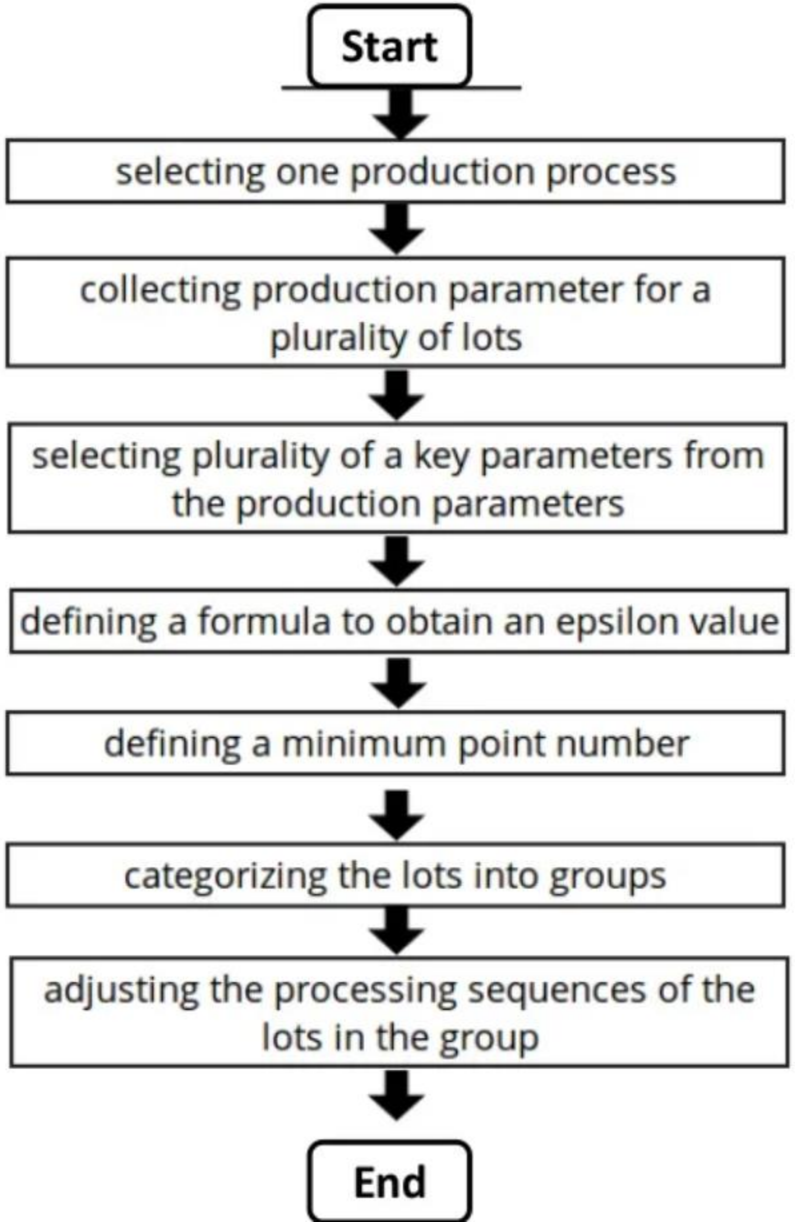
Input: N objects to be clustered and global parameters *Eps*, *MinPts*.

Output: Clusters of objects.

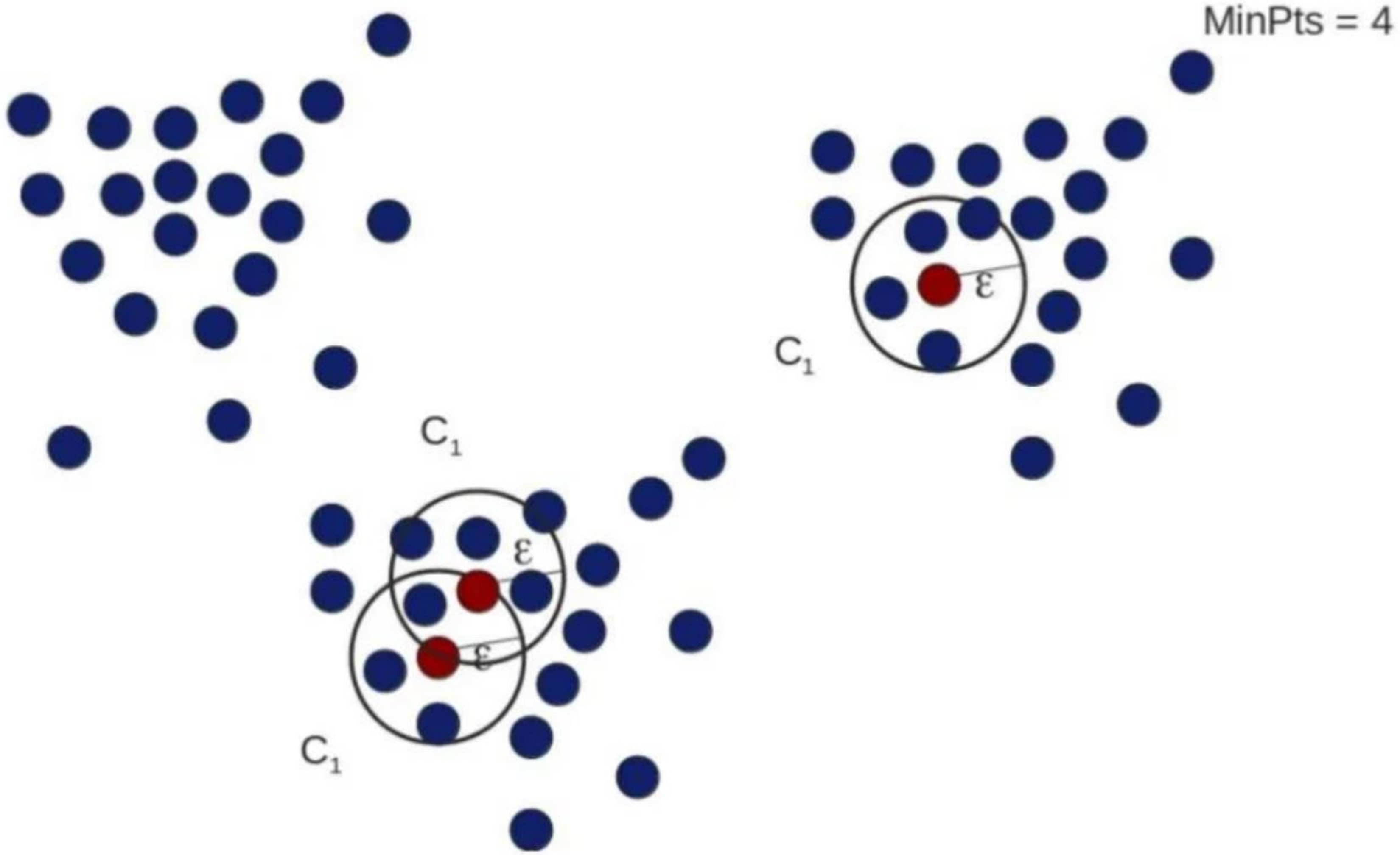
Algorithm:

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

DBScan :Flowchart



DBScan : Example



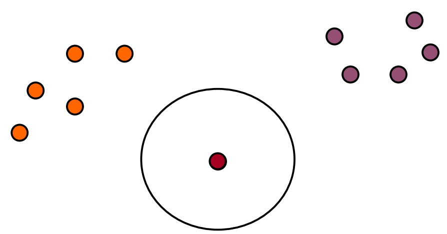
Summary of DBSCAN

Good:

- can detect arbitrary shapes,
- not very sensitive to noise,
- supports outlier detection,
- complexity is kind of okay,
- beside K-means the second most used clustering algorithm.

DBSCAN Algorithm: Example

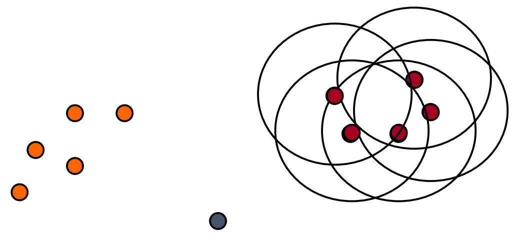
- Parameter
  - ε = 2 cm
  - *MinPts* = 3



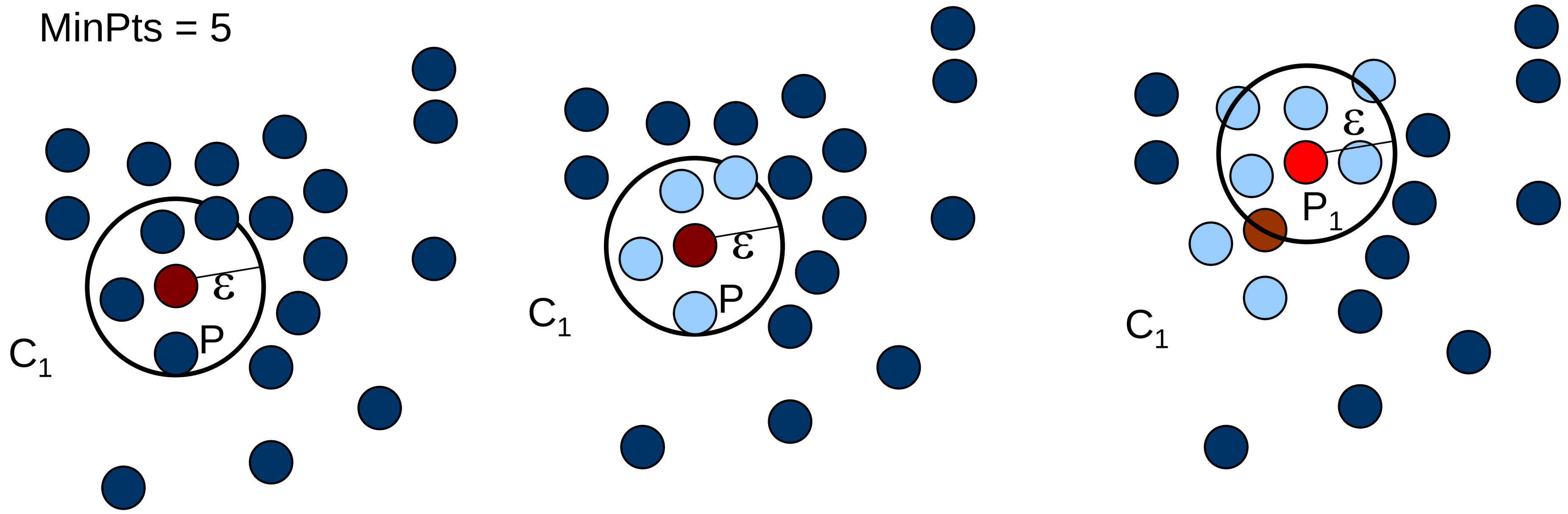
for each $o \in D$ **do**
    if o is not yet classified **then**
        if o is a core-object **then**
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE

DBSCAN Algorithm: Example

- Parameter
  - ε = 2 cm
  - *MinPts* = 3

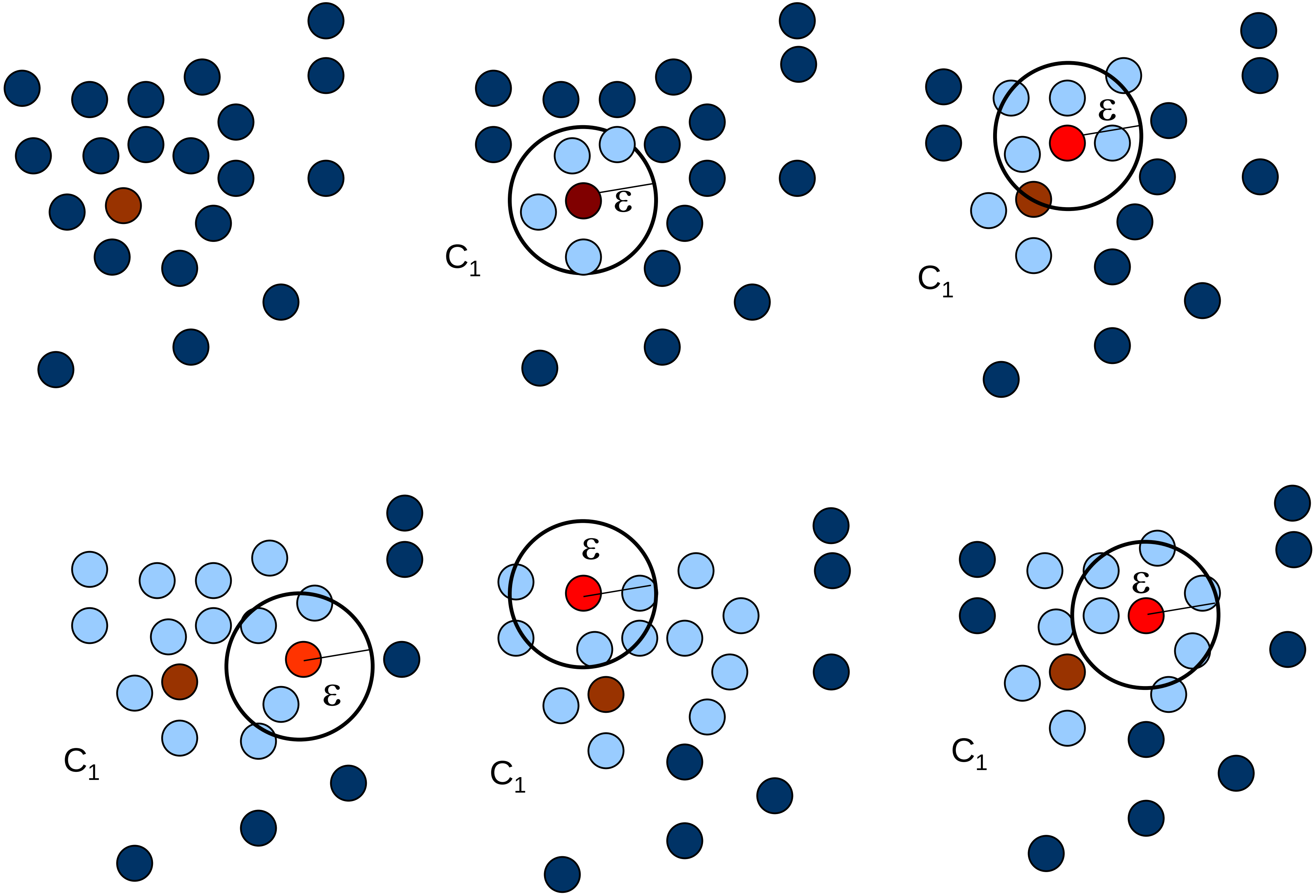


for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE



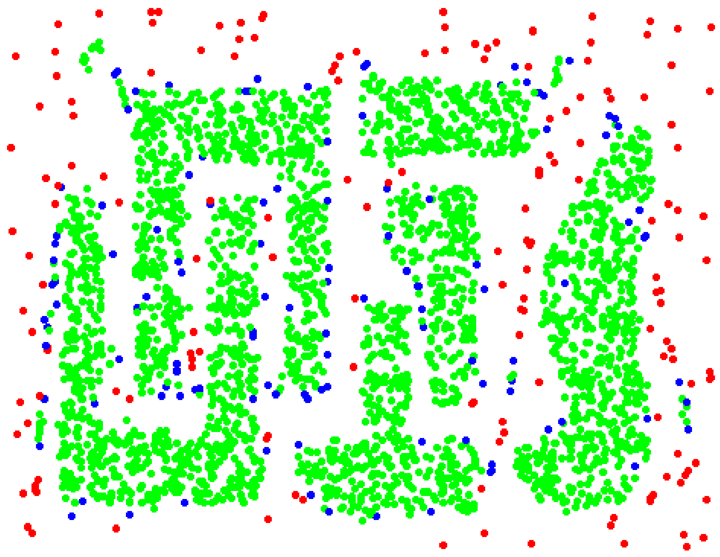
1. Check the ε-neighborhood of p;
2. If p has less than MinPts neighbors then mark p as outlier and continue with the next object
3. Otherwise mark p as processed and put all the neighbors in cluster C

1. Check the unprocessed objects in C
2. If no core object, return C
3. Otherwise, randomly pick up one core object p_1, mark p_1 as processed, and put all unprocessed neighbors of p_1 in cluster C


ε
C_1
ε
C_1
ε
C_1
ε
C_1
ε
C_1

Example

Original Points

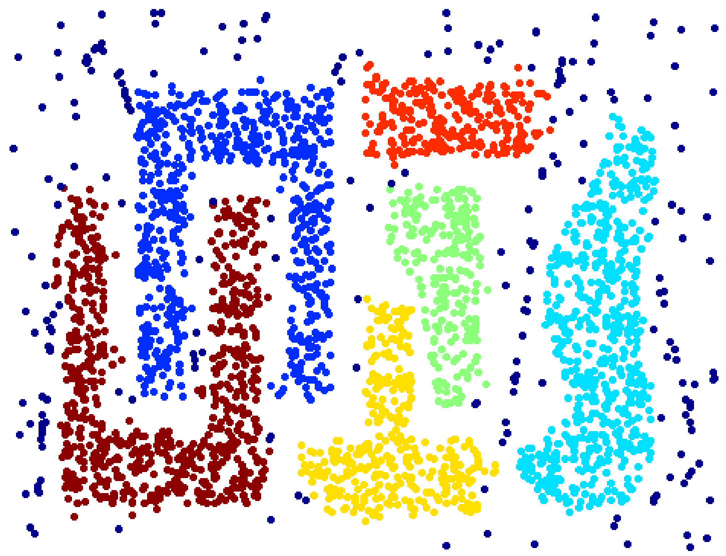


Point types: core, border and outliers

ε = 10, MinPts = 4

When DBSCAN Works Well

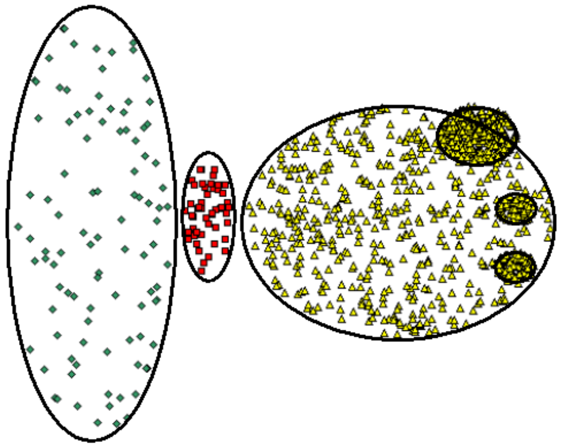
Original Points



Clusters

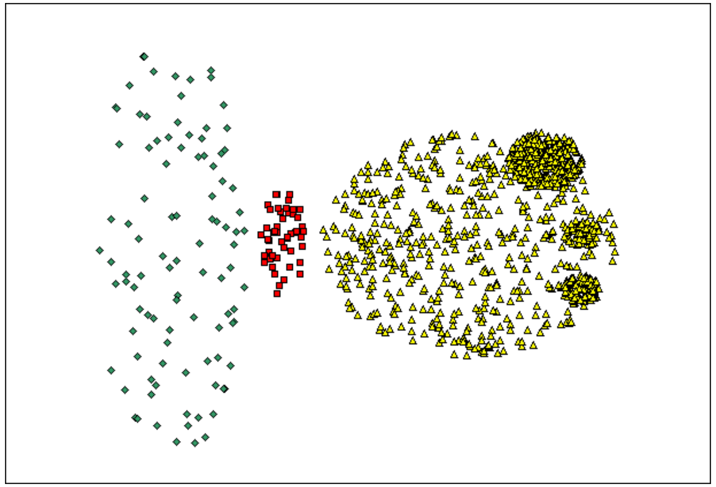
- Resistant to Noise
- Can handle clusters of different shapes and sizes

When DBSCAN Does NOT Work Well

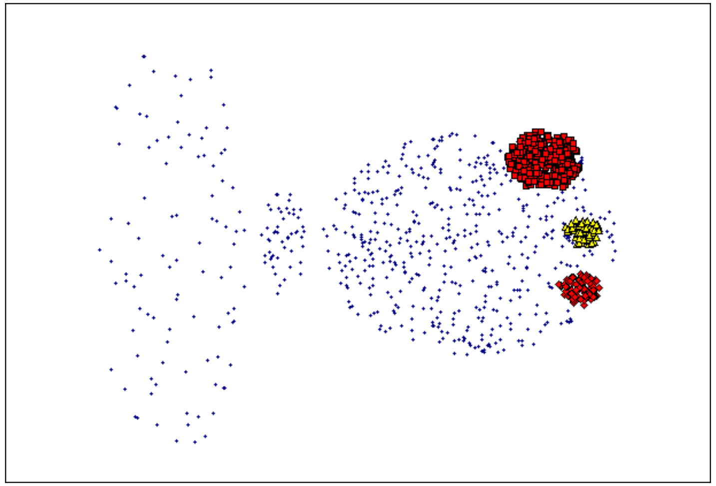


Original Points

- **Cannot handle Varying densities**
- **sensitive to parameters**



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

DBSCAN: Sensitive to Parameters

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

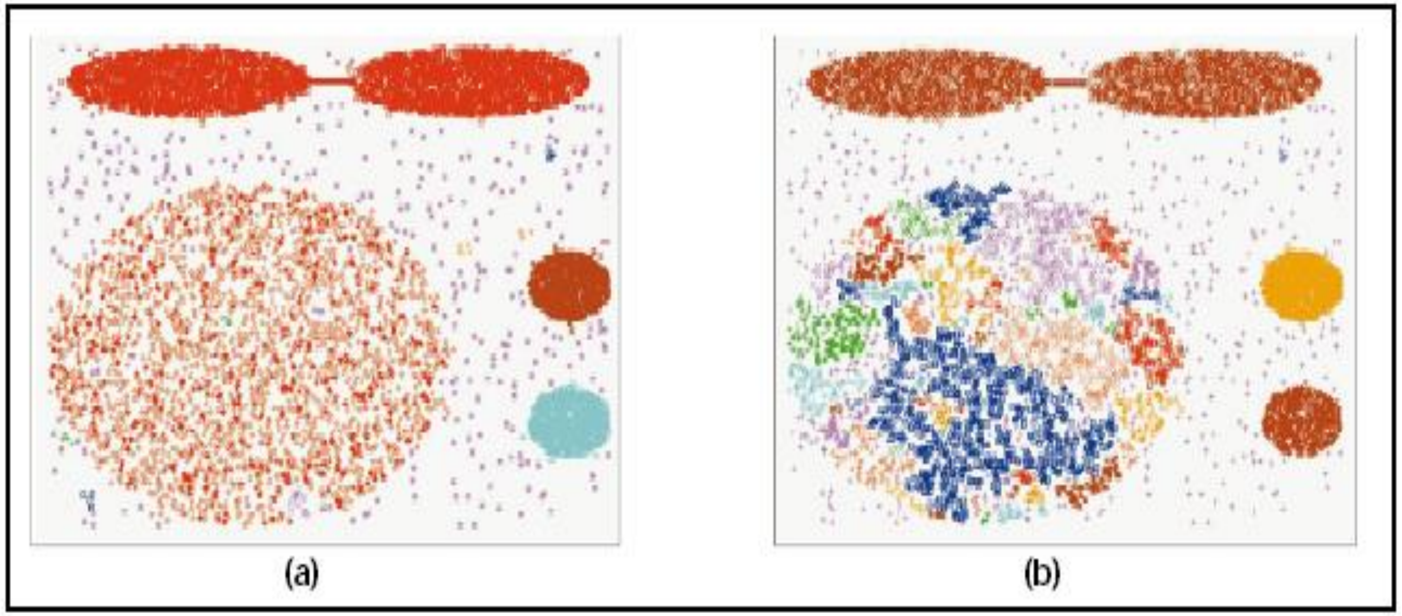
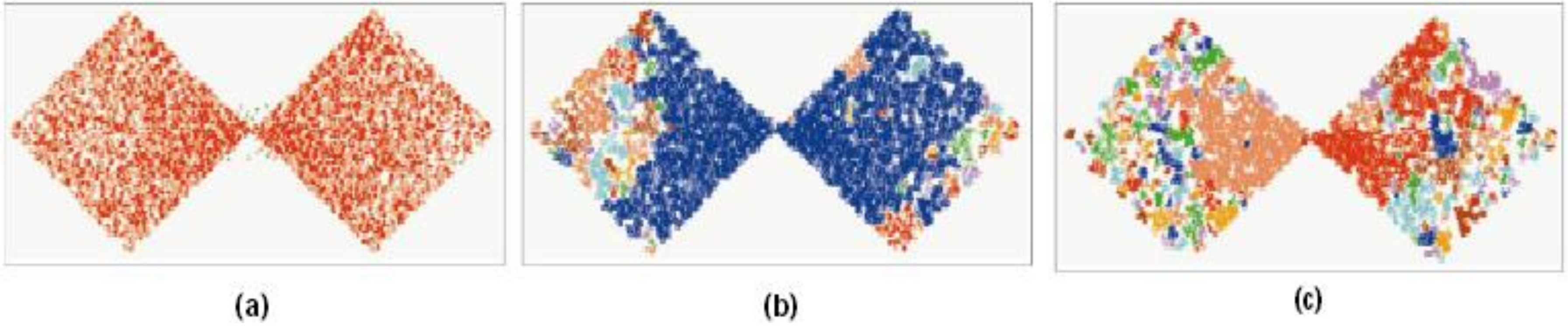


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



Agenda

- Association
  - Apriori
  - Market Basket Analysis

`Basket data’

A very common type of data; often also called *transaction data*.

Next slide shows example *transaction database*, where each record represents a transaction between (usually) a customer and a shop.

Each record in a supermarket’s transaction DB, for example, corresponds to a basket of specific items.

ID	apples,	beer,	cheese,	dates,	eggs,	fish,	glue,	honey,	ice-cream
1	1	1		1			1	1	
2			1	1	1				
3		1	1			1			
4		1				1			1
5					1		1		
6						1			1
7	1			1				1	
8						1			1
9			1		1				
10		1					1		
11					1		1		
12	1								
13			1			1			
14			1			1			
15								1	1
16				1					
17	1					1			
18	1	1	1	1				1	
19	1	1		1			1	1	
20					1				

Data

Transactions	*Items*
1	A B C D
2	A C D
3	A B C
4	C D E
5	A B C E

Matrix representation

Transactions \ *Items*	A	B	C	D	E
1	1	1	1	1	0
2	1	0	1	1	0
3	1	1	1	0	0
4	0	0	1	1	1
5	1	1	1	0	1

Execution of Apriori algorithm, $\varepsilon = 2$

Iteration 1

Candidates of size 1	*Support*
A	4
B	3
C	5
D	3
~~E~~	~~1~~

Iteration 2

Candidates of size 2	*Support*
A B	3
A C	4
A D	2
B C	3
~~B D~~	~~1~~
C D	3

Iteration 3

Candidates of size 3	*Support*
A B C	3
~~A B D~~	~~1~~
A C D	2

$$Rule:\ X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

$$Support = \frac{P(A \cap B)}{n}$$

$$Confidence = \frac{P(A \cap B)}{P(A)}$$

$$Lift = \frac{P(A \cap B)}{P(A) \bullet P(B)}$$

Support ($A \Rightarrow B$) = $P(A \cap B)$
Confidence ($A \Rightarrow B$) =$P(B|A)$
Lift ($A \Rightarrow B$) = $P(B|A)/P(B)$

Discovering Rules

A common and useful application of data mining

A `rule' is something like this:

If a basket contains apples and cheese, then it also contains beer

Any such rule has two associated measures:

1. *confidence* – when the `if' part is true, how often is the `then' bit true? This is the same as *accuracy*.

2. *coverage* or *support* – how much of the database contains the `if' part?

Apriori Algorithm – Iteration 4

Prune the itemset as its 3-item subset is not frequent in L3

Itemset	Supp Count
{I1,I2,I3}	2
{I1,I2,I5}	2

L3

Generate Candidate 4-Itemsets by joining L3 X L3

Itemset
{I1,I2,I3,I5}

C4

After Pruning

Itemset
Empty

Stop

Generating Association Rule Example

Frequent Itemset – {I1,I2,I5} Minimum Confidence = 70%

Association Rules

{I1,I2} => I5	Confidence = 2/4 = 50 %	✗
{I1,I5} => I2	Confidence = 2/2 = 100 %	✓
{I2,I5} => I1	Confidence = 2/2 = 100 %	✓
I1 => {I2,I5}	Confidence = 2/6 = 33 %	✗
I2 => {I1,I5}	Confidence = 2/7 = 29 %	✗
I5 => {I1,I2}	Confidence = 2/2 = 100 %	✓

Item set	Sup-count
Hot Dogs	4
Buns	2
Ketchup	2
Coke	3
Chips	4

Item set	Sup-count
Hot Dogs	4
Buns	2
Ketchup	2
Coke	3
Chips	4

Item set	Sup-count
Hot Dogs, Buns	2
Hot Dogs, Ketchup	1
Hot Dogs, Coke	2
Hot Dogs, Chips	2
Buns, Ketchup	1
Buns, Coke	0
Buns, Chips	0
Ketchup, Coke	0
Ketchup, Chips	1
Coke, Chips	3

Item set	Sup-count
Hot Dogs, Buns	2
Hot Dogs, Coke	2
Hot Dogs, Chips	2
Coke, Chips	3

Item set	Sup-count
Hot Dogs, Buns, Coke	0
Hot Dogs, Buns, Chips	0
Hot Dogs, Coke, Chips	2

Item set	Sup-count
Hot Dogs, Coke, Chips	2

Example:

What is the confidence and coverage of:

If the basket contains beer and cheese, then it also contains honey

2/20 of the records contain both beer and cheese, so coverage is 10%

Of these 2, 1 contains honey, so confidence is 50%

Interesting means surprising

We therefore have a prior expectation that just 4 in 1,000 baskets should contain **both** bread and washing up powder.

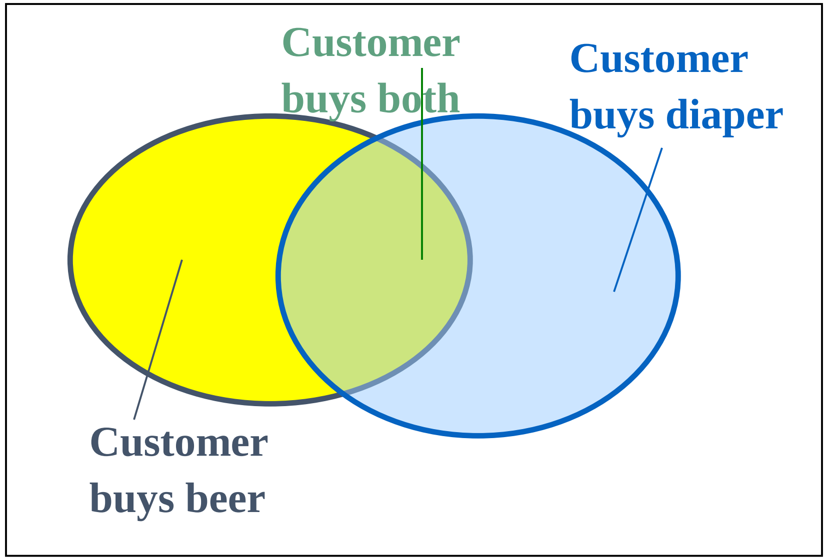
If we investigate, and discover that really it is 20 in 1,000 baskets, then we will be very surprised. It tells us that:

- Something is going on in shoppers' minds: bread and washing-up powder are connected in some way.
- There may be ways to exploit this discovery … put the powder and bread at opposite ends of the supermarket?

Measure	Description	Formula
Support	The usefulness of discovered rule $A \rightarrow B$	$P(A \cap B)$
Confidence	The certainty of discovered rule $A \rightarrow B$	$P(B \mid A)$
Lift	The correlation between the occurrence of items in discovered rule $A \rightarrow B$.	$\dfrac{P(B \mid A)}{P(B)}$

Basic Concepts: Frequent Patterns

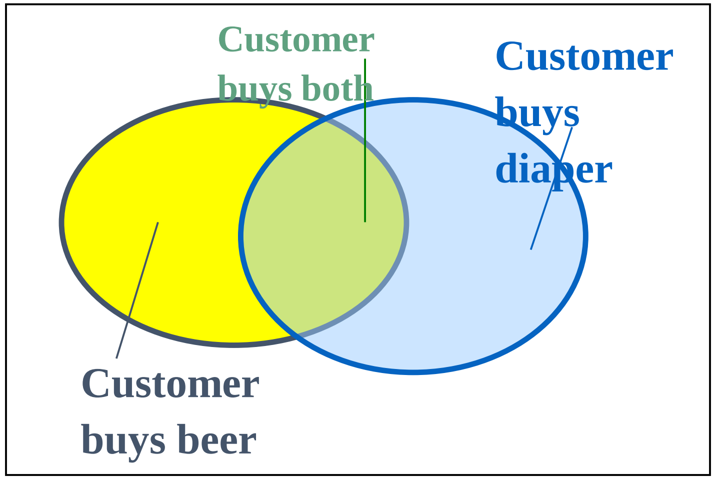
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- itemset: A set of one or more items
- k-itemset $X = \{x_1, \ldots, x_k\}$
- *(absolute) support*, or, *support count* of X: Frequency or occurrence of an itemset X
- *(relative) support*, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is *frequent* if X's support is no less than a *minsup* threshold

Basic Concepts: Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, s, probability that a transaction contains $X \cup Y$
  - confidence, c, conditional probability that a transaction having X also contains Y

Let minsup = 50%, minconf = 50%

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- Association rules: (many more!)
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)