

Density Based Clustering DBSCAN

By
R. Siva Narayana

DBSCAN

Density Based Spatial Clustering of Applications with Noise

$N_{Eps}(p)$: { belongs to **D** | $\text{dist}(p, q) \leq Eps, q = 1 \text{ to } n$ }

Directly density-reachable:

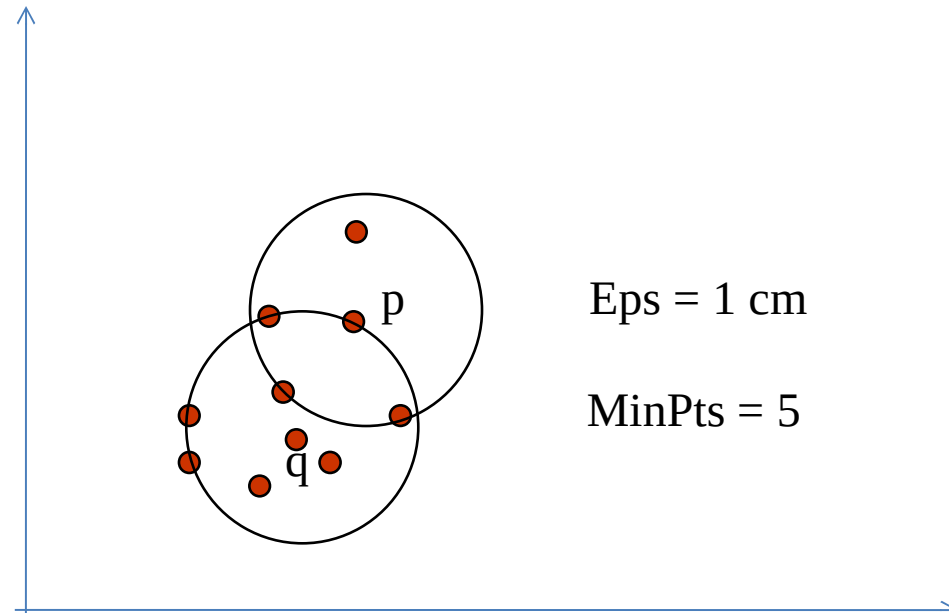
A point p is directly density-reachable from a point q and q must be core object

w.r.t. $Eps, MinPts$ if

p belongs to $N_{Eps}(q)$

core condition:

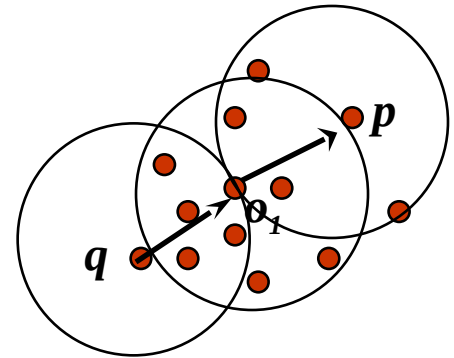
$|N_{Eps}(q)| \geq MinPts$



DBSCAN

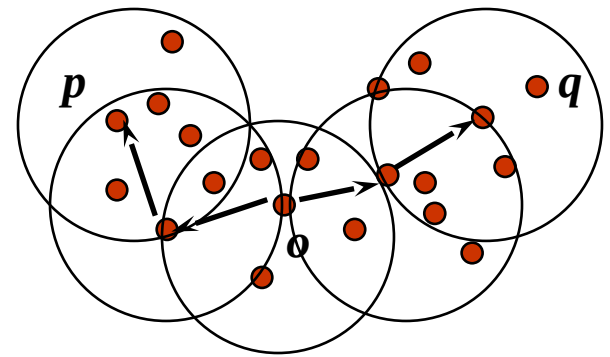
- **Density-reachable:**

- A point p is **density-reachable** from a point q w.r.t. Eps , $MinPts$ if there is a chain of points $o_1, \dots, o_n$, $o_1 = q$, $o_n = p$ such that o_i is directly density-reachable from o_{i+1}



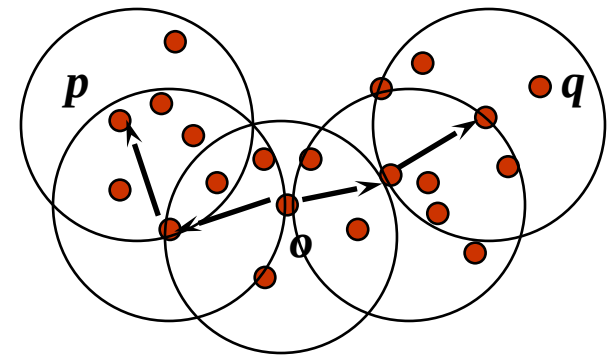
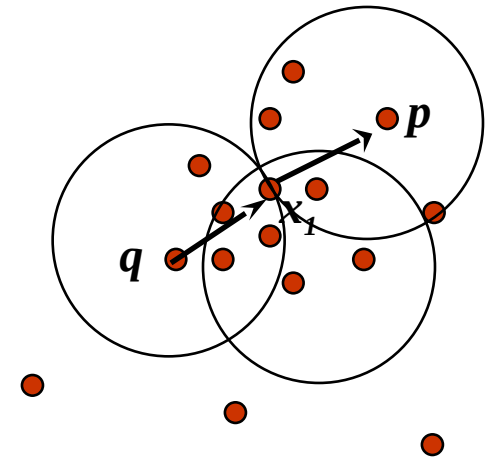
- **Density-connected:**

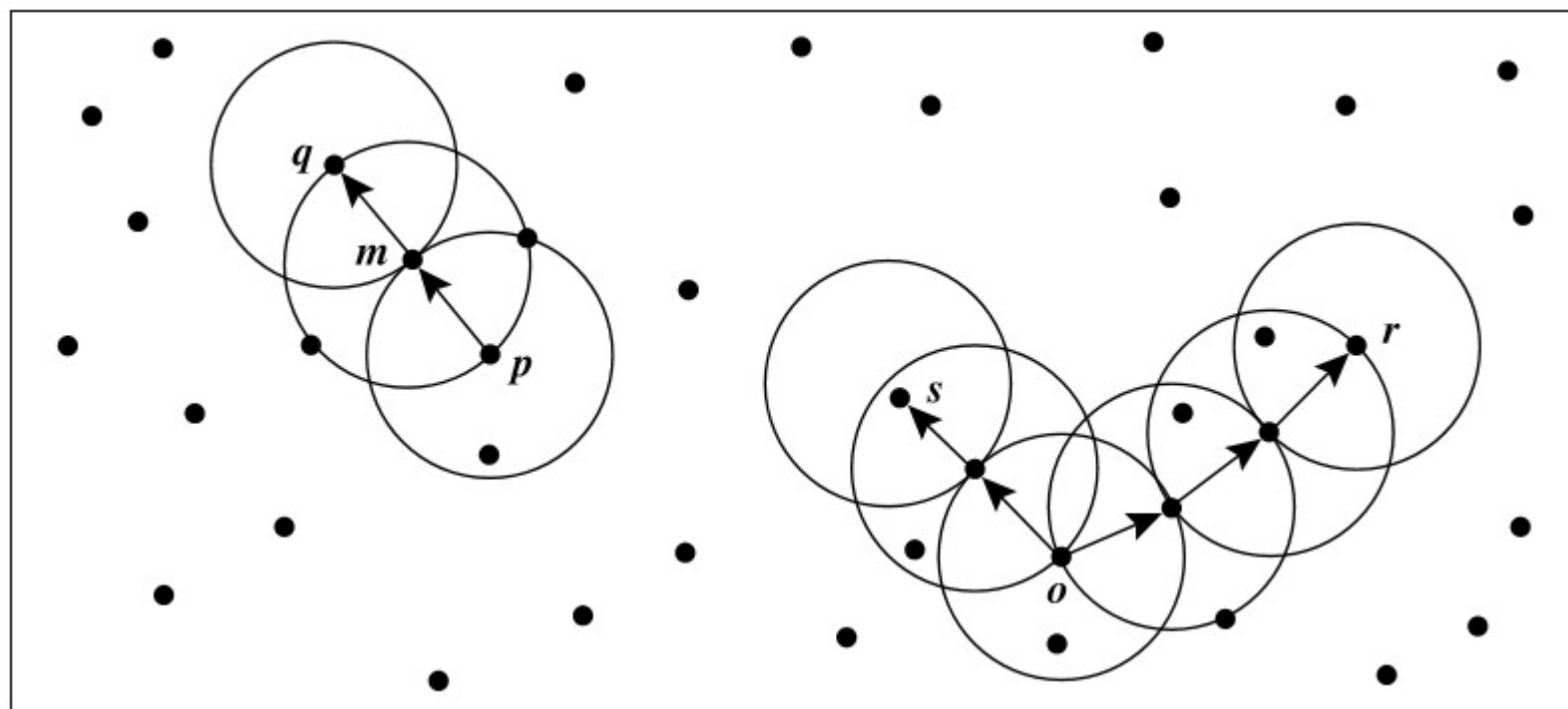
- A point p is **density-connected** to a point q w.r.t. Eps , $MinPts$ if there is a point o such that both, p and q are density-reachable from o w.r.t. Eps and $MinPts$



DBSCAN

- **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





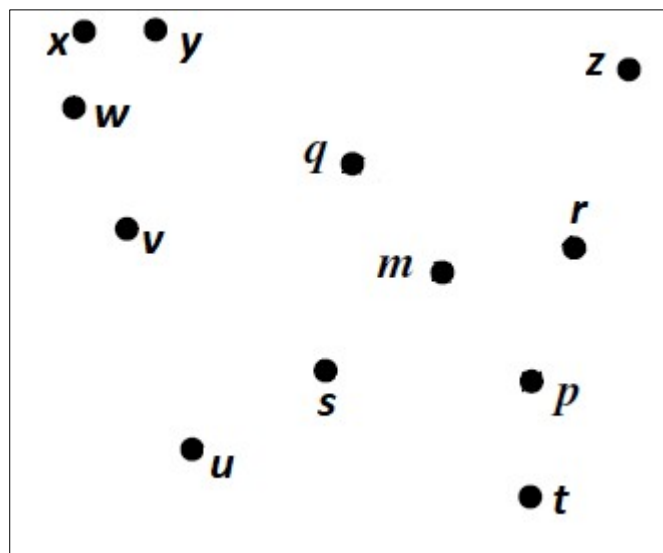
ALGORITHM DBSCAN

Algorithm: DBSCAN: a density-based clustering algorithm.

Input:

- D : a data set containing n objects,
- ϵ : the radius parameter, and
- $MinPts$: the neighborhood density threshold.

Output: A set of density-based clusters.



ALGORITHM DBSCAN

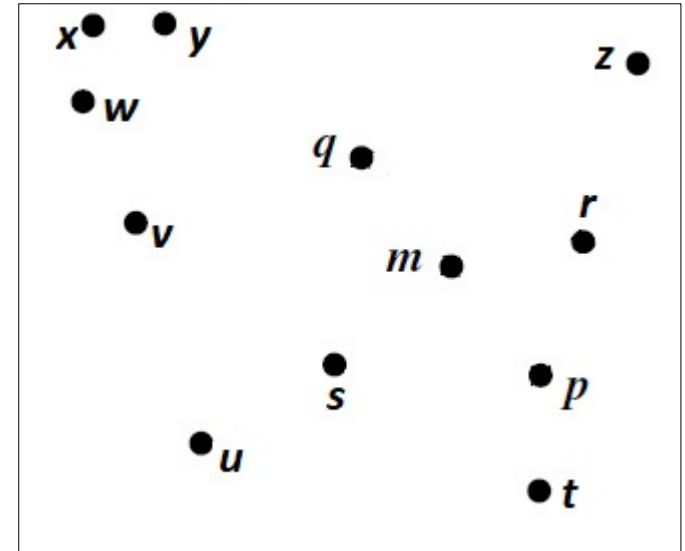
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object $\mathbf{O}$;
- (4) mark $\mathbf{O}$ as visited;
- (5) **if** the ϵ -neighborhood of $\mathbf{O}$ has at least *MinPts* objects
- (6) create a new cluster C , and add $\mathbf{O}$ to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of $\mathbf{O}$;
- (8) **for** each point $\mathbf{O}'$ in N
- (9) **if** $\mathbf{O}'$ is unvisited
- (10) mark $\mathbf{O}'$ as visited;
- (11) **if** the ϵ -neighborhood of $\mathbf{O}'$ has at least *MinPts* points,
 add those points to N ;
- (12) **if** $\mathbf{O}'$ is not yet a member of any cluster, add $\mathbf{O}'$ to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark $\mathbf{O}$ as noise;
- (16) **until** no object is unvisited;

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Example



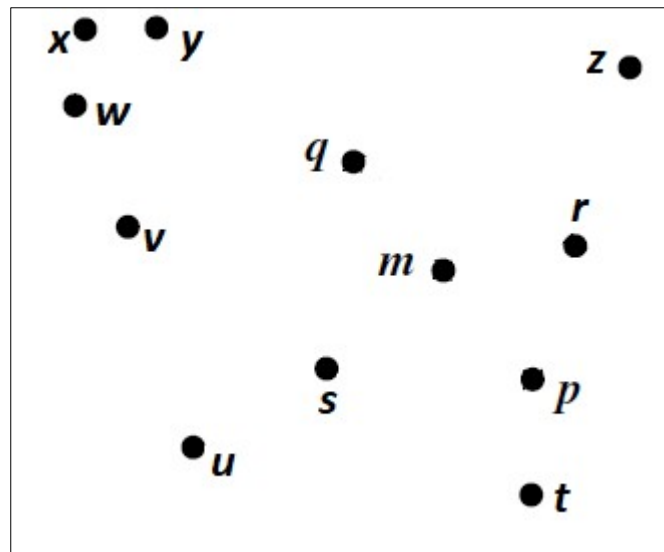
$\epsilon = 1 \text{ cm}$
MinPts = 4

Unvisited **m p q r s t u v w x y z**

↑

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least $MinPts$ objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least $MinPts$ points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



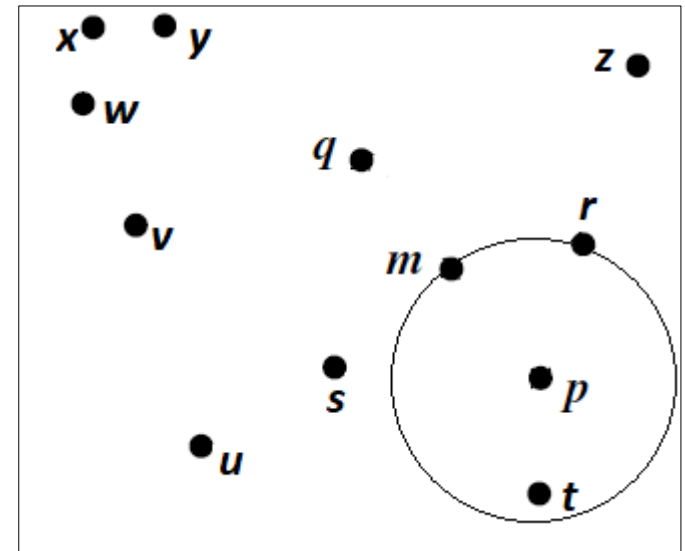
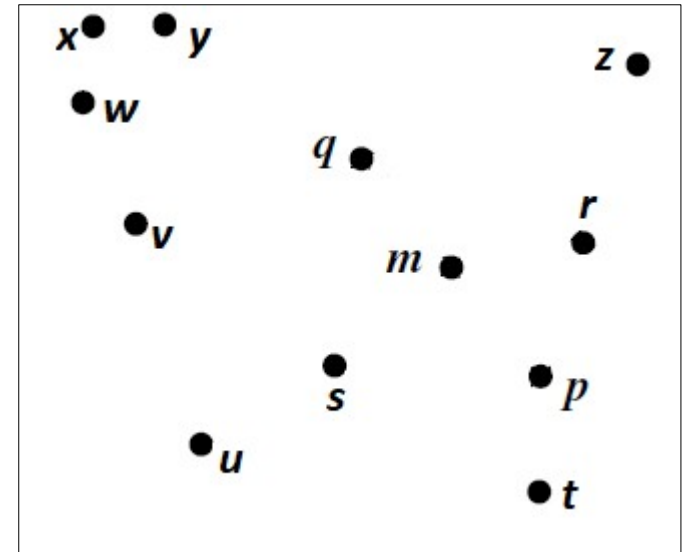
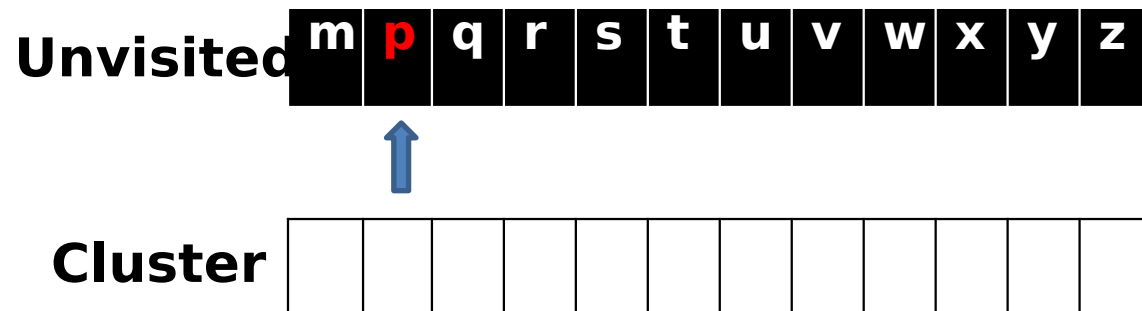
Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

↑

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

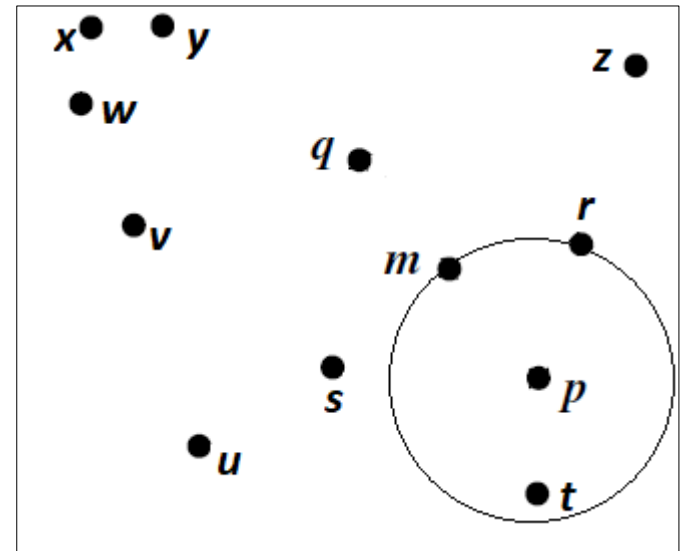
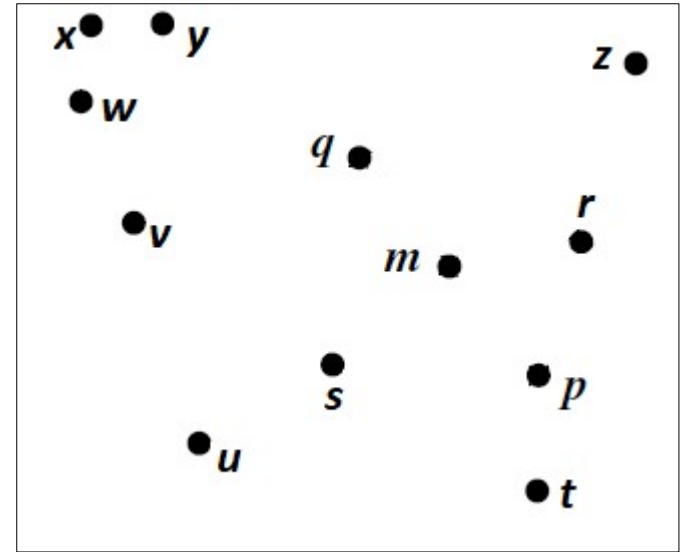


Method:

- ```

(1) mark all objects as unvisited;
(2) do
(3) randomly select an unvisited object O ;
(4) mark O as visited;
(5) if the ϵ -neighborhood of O has at least $MinPts$ objects
(6) create a new cluster C , and add O to C ;
(7) let N be the set of objects in the ϵ -neighborhood of O ;
(8) for each point O' in N
(9) if O' is unvisited
(10) mark O' as visited;
(11) if the ϵ -neighborhood of O' has at least $MinPts$ points,
 add those points to N ;
(12) if O' is not yet a member of any cluster, add O' to C ;
(13) end for
(14) output C ;
(15) else mark O as noise;
(16) until no object is unvisited;

```



Unvisited

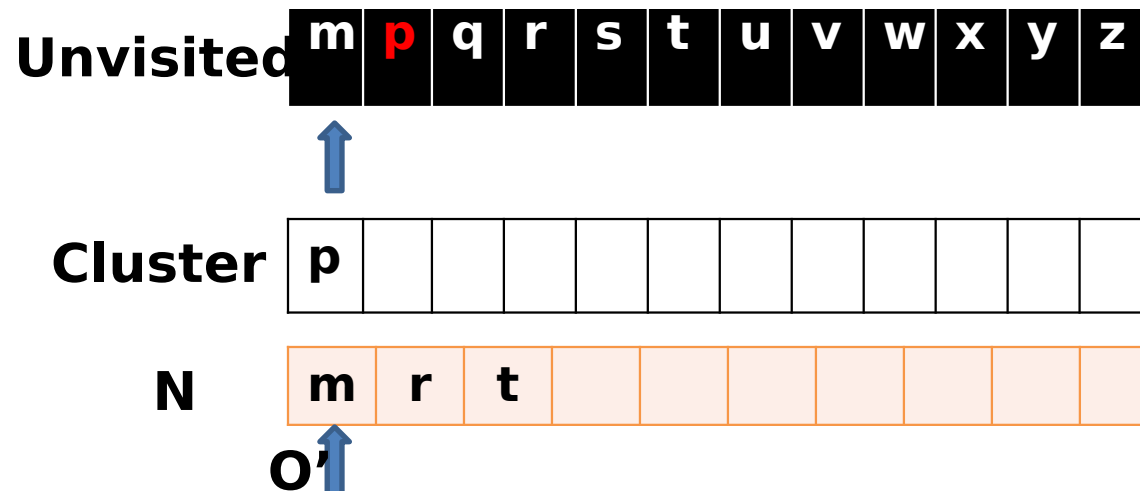
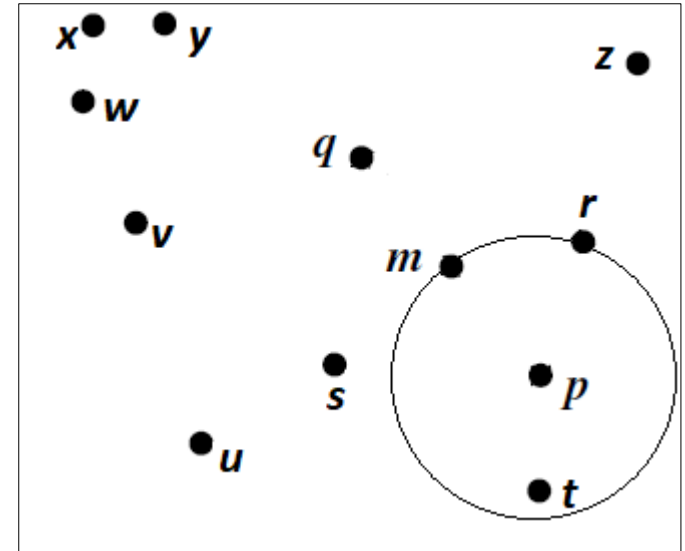
Cluster

N

O

### Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3)     randomly select an unvisited object  $O$ ;
- (4)     mark  $O$  as visited;
- (5)     **if** the  $\epsilon$ -neighborhood of  $O$  has at least MinPts objects
- (6)         create a new cluster  $C$ , and add  $O$  to  $C$ ;
- (7)         let  $N$  be the set of objects in the  $\epsilon$ -neighborhood of  $O$ ;
- (8)         **for** each point  $O'$  in  $N$
- (9)             **if**  $O'$  is unvisited
- (10)                 mark  $O'$  as visited;
- (11)                 **if** the  $\epsilon$ -neighborhood of  $O'$  has at least MinPts points,  
                    add those points to  $N$ ;
- (12)             **if**  $O'$  is not yet a member of any cluster, add  $O'$  to  $C$ ;
- (13)         **end for**
- (14)         output  $C$ ;
- (15)     **else** mark  $O$  as noise;
- (16) **until** no object is unvisited;

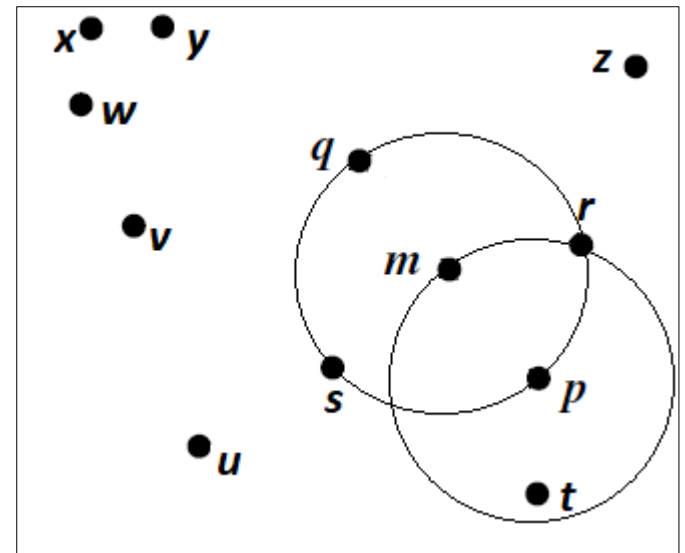
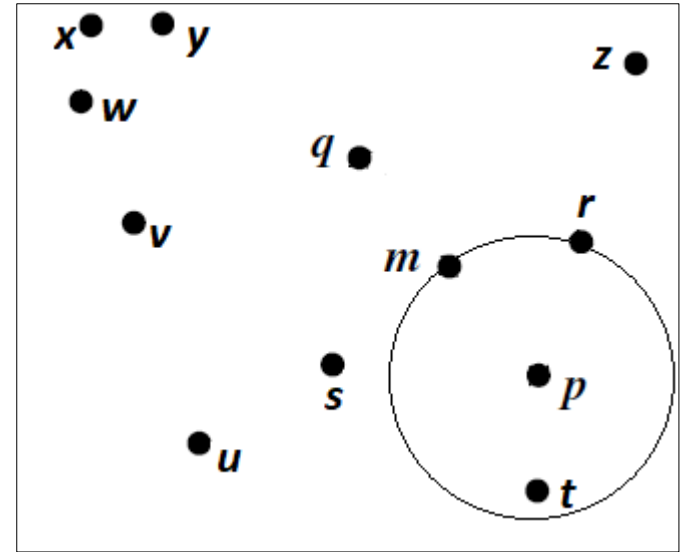


### Method:

- ```

(1) mark all objects as unvisited;
(2) do
(3)     randomly select an unvisited object  $O$ ;
(4)     mark  $O$  as visited;
(5)     if the  $\epsilon$ -neighborhood of  $O$  has at least  $MinPts$  objects
(6)         create a new cluster  $C$ , and add  $O$  to  $C$ ;
(7)         let  $N$  be the set of objects in the  $\epsilon$ -neighborhood of  $O$ ;
(8)         for each point  $O'$  in  $N$ 
(9)             if  $O'$  is unvisited
(10)                mark  $O'$  as visited;
(11)                if the  $\epsilon$ -neighborhood of  $O'$  has at least  $MinPts$  points,
                    add those points to  $N$ ;
(12)                if  $O'$  is not yet a member of any cluster, add  $O'$  to  $C$ ;
(13)         end for
(14)     output  $C$ ;
(15) else mark  $O$  as noise;
(16) until no object is unvisited;

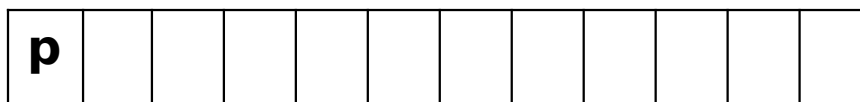
```



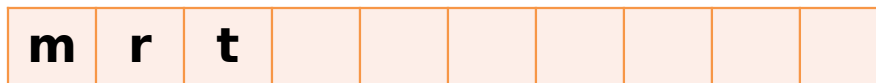
Unvisited **m** **p** q r s t u v w x y z



Cluster



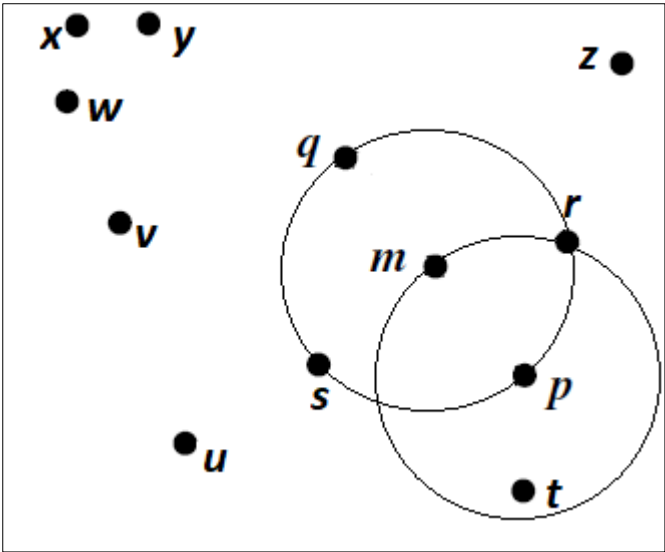
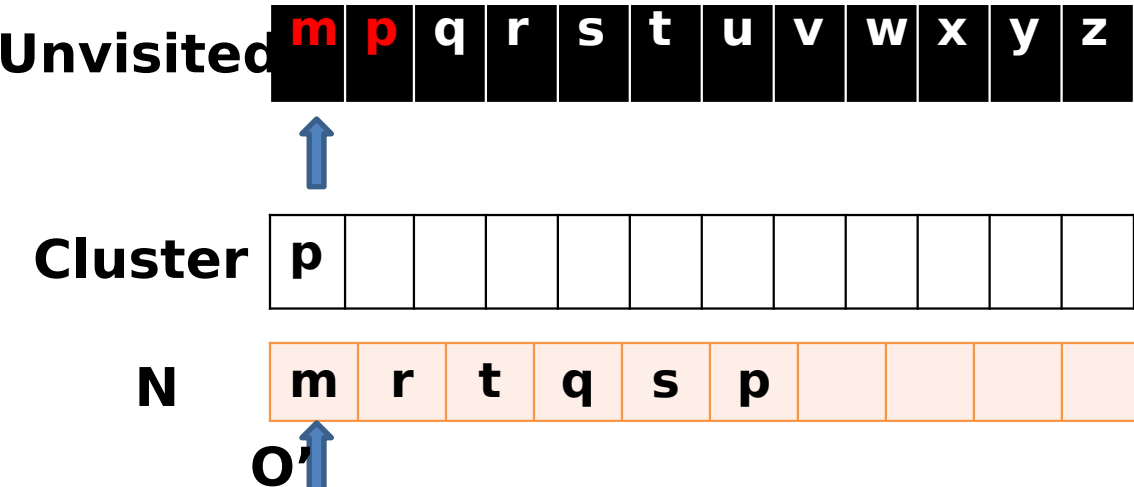
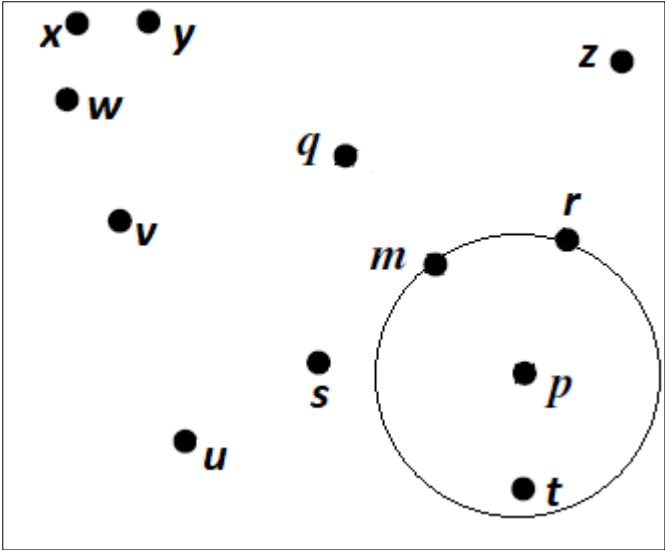
N



o'

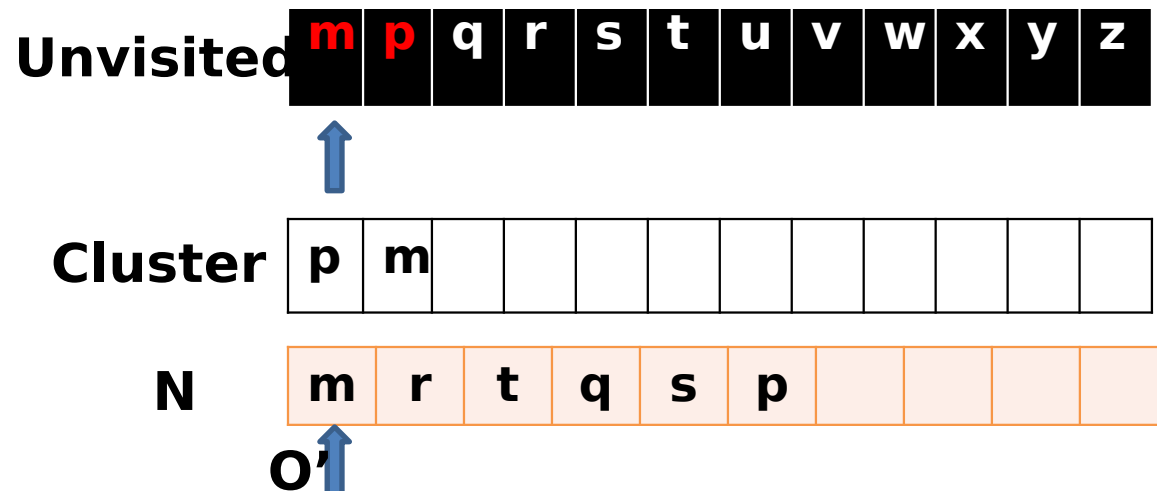
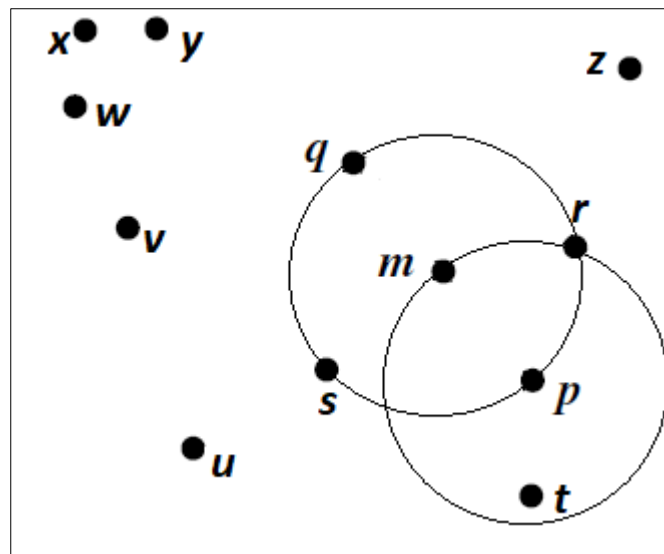
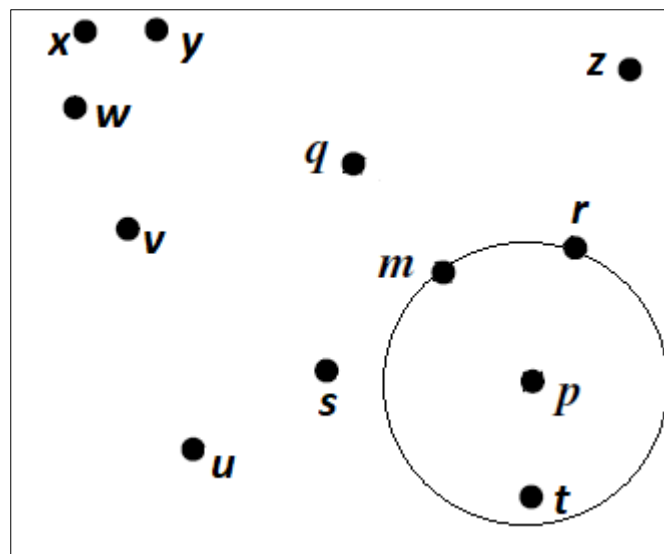
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



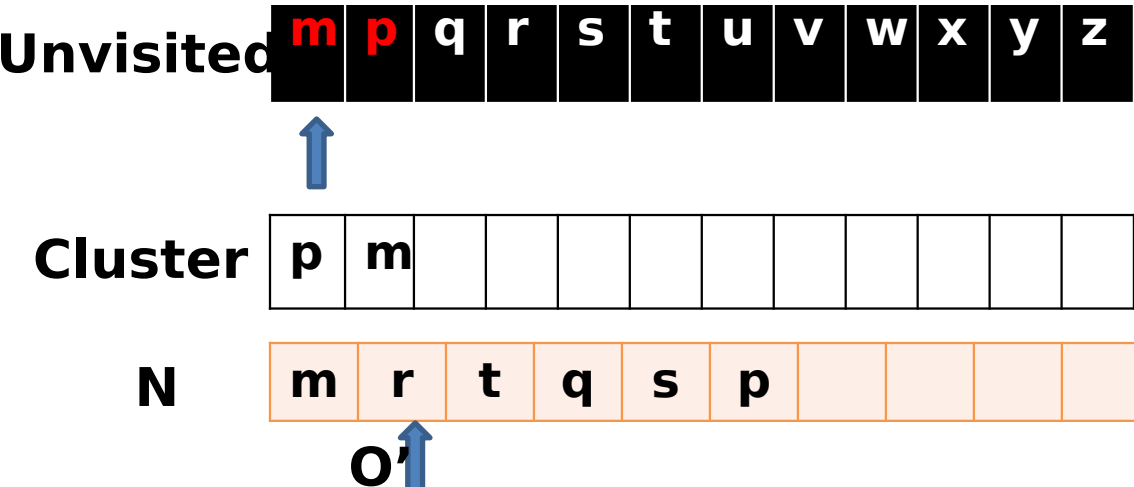
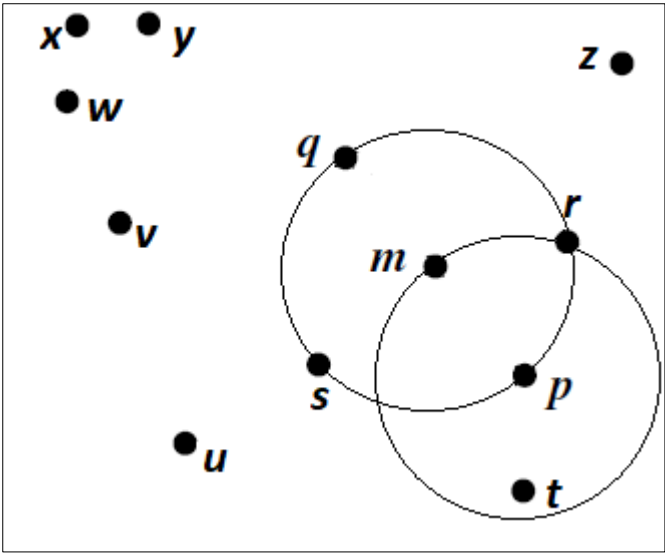
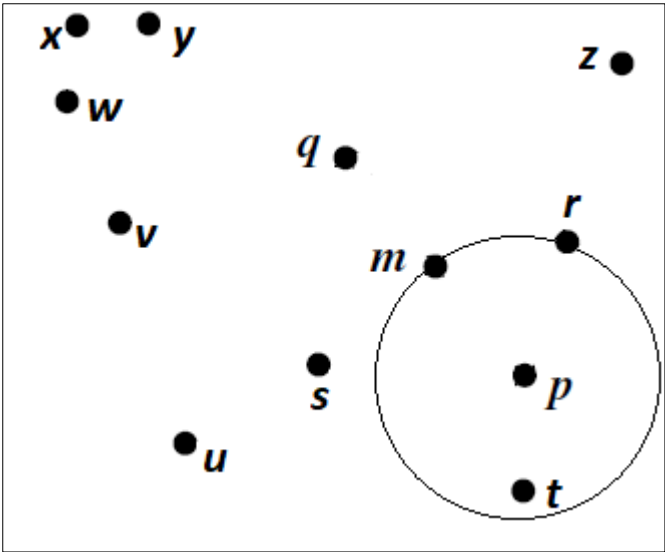
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



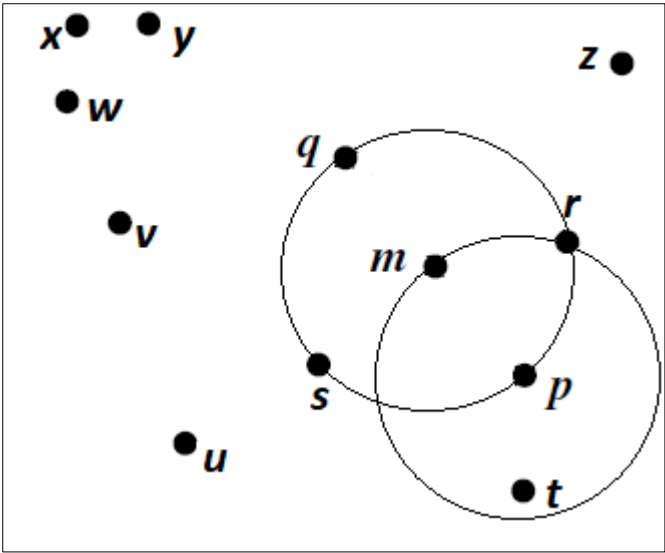
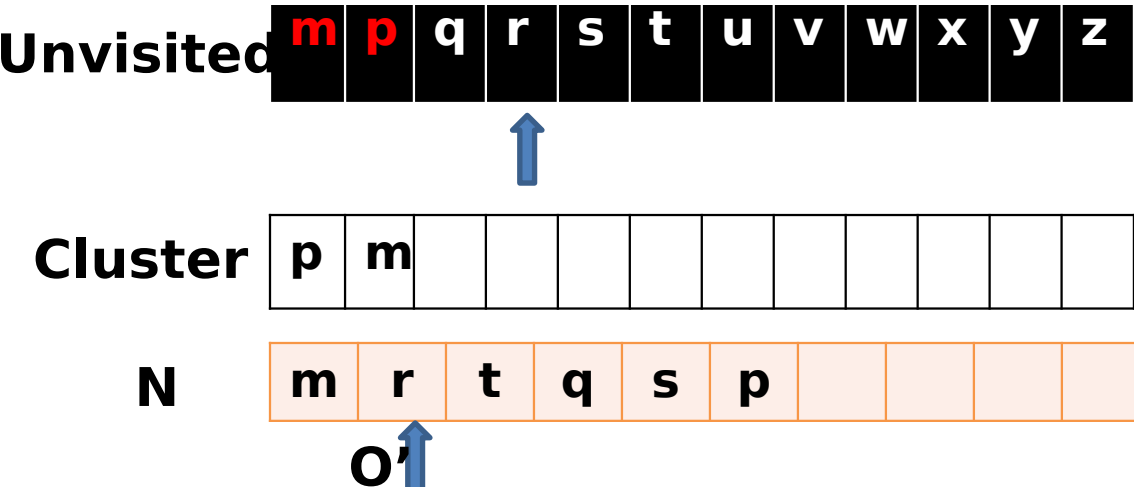
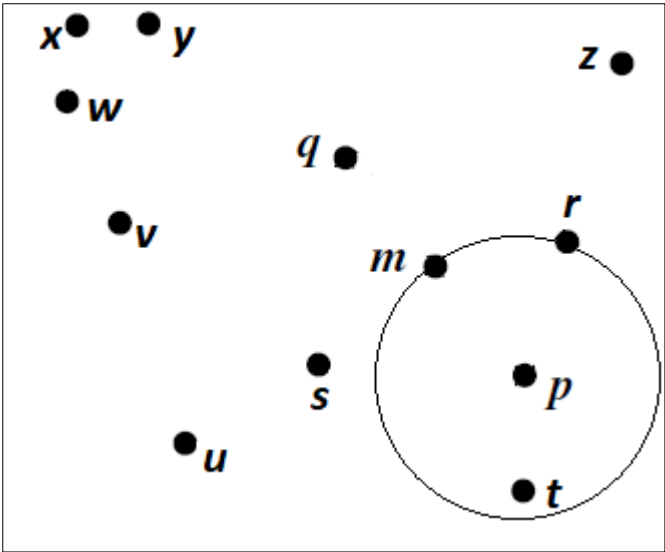
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



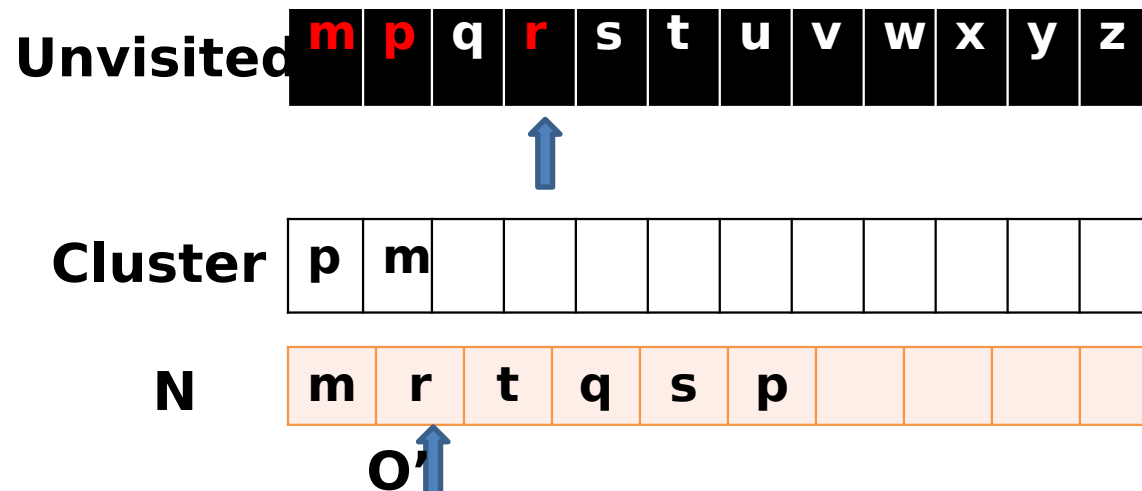
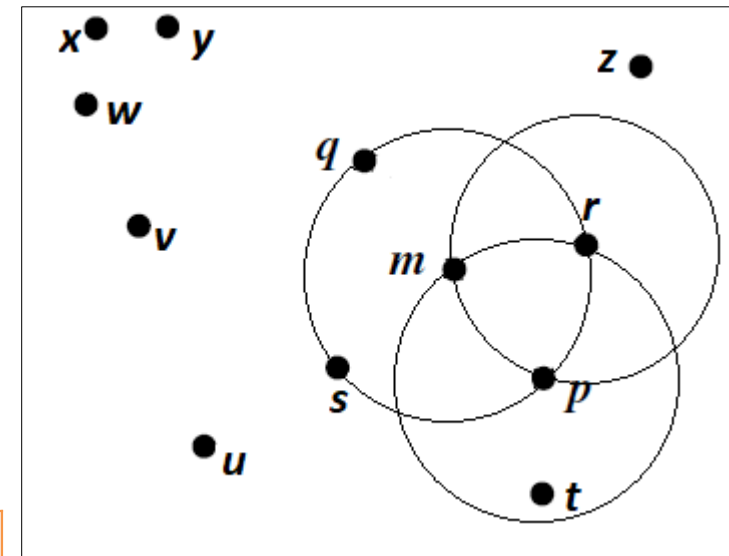
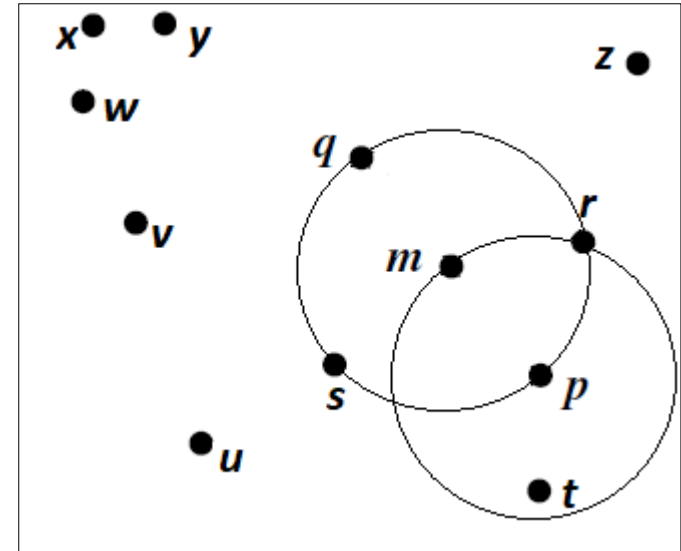
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



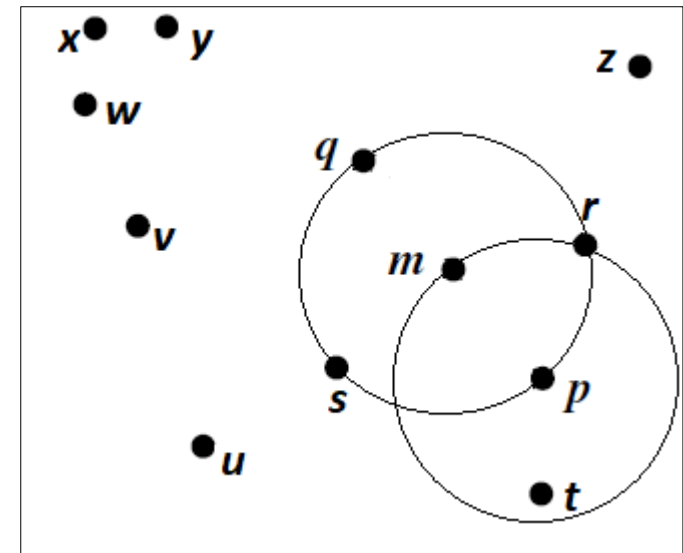
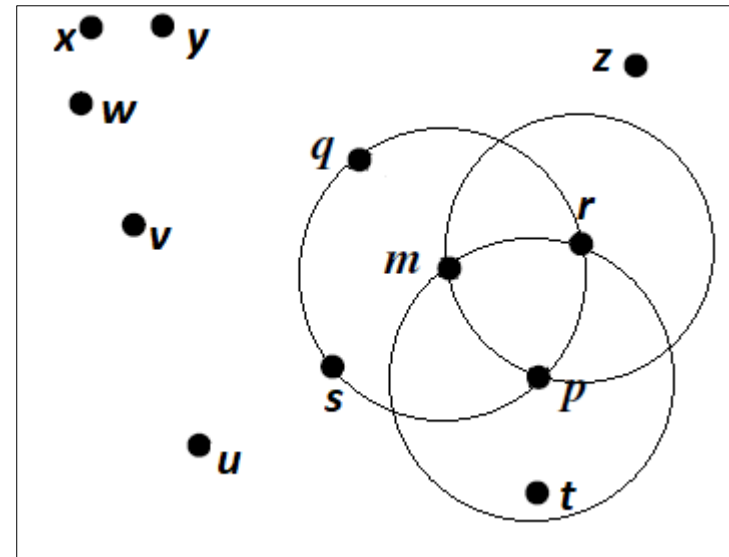
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited **m** **p** **q** **r** s t u v w x y z



Cluster

p	m	r									
---	---	---	--	--	--	--	--	--	--	--	--

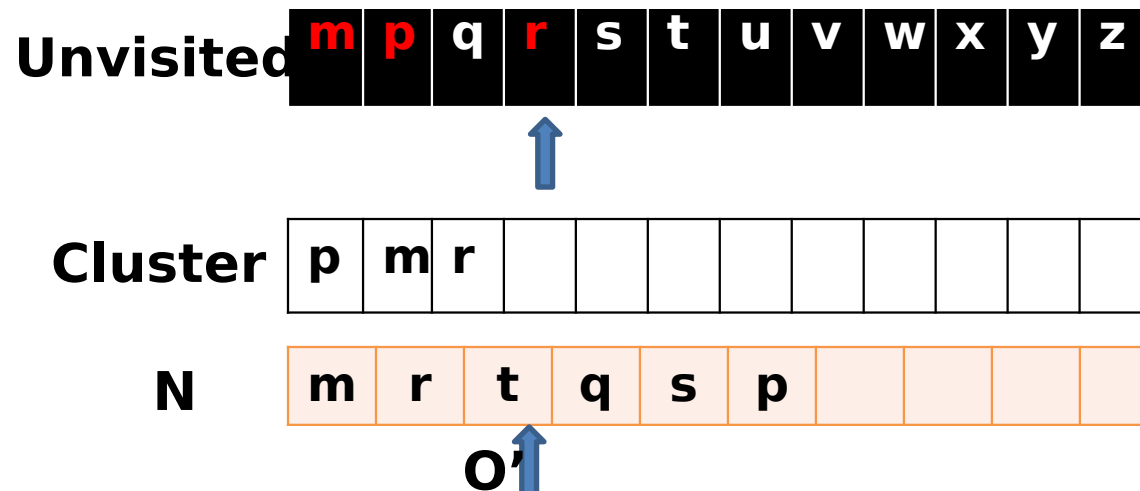
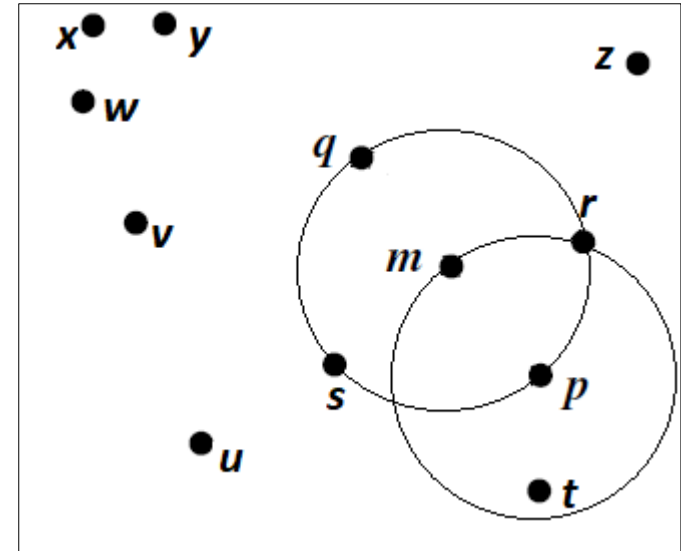
N

m	r	t	q	s	p						
---	---	---	---	---	---	--	--	--	--	--	--



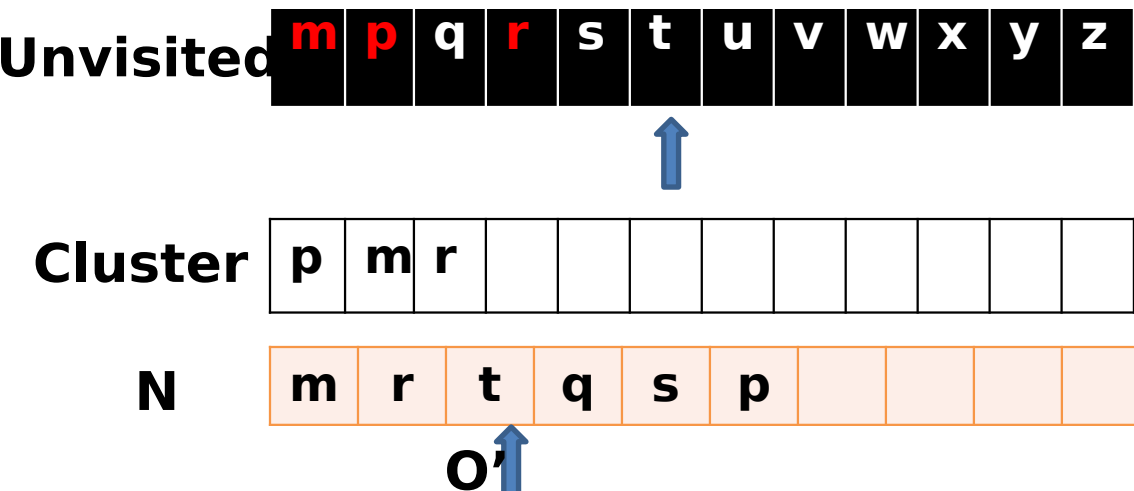
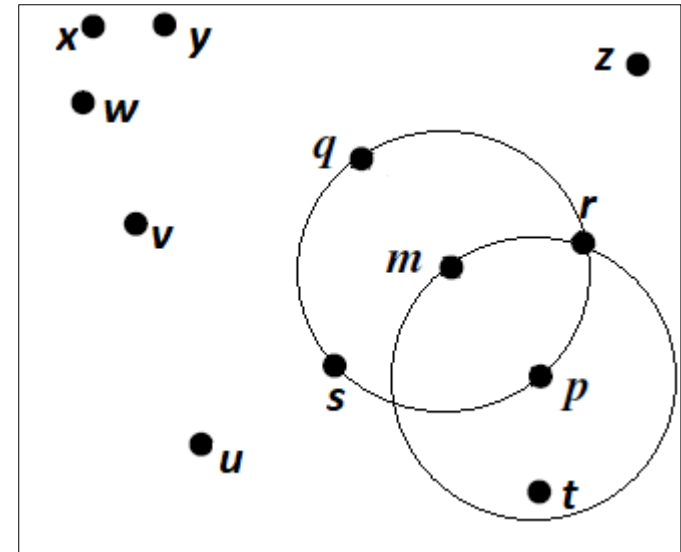
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



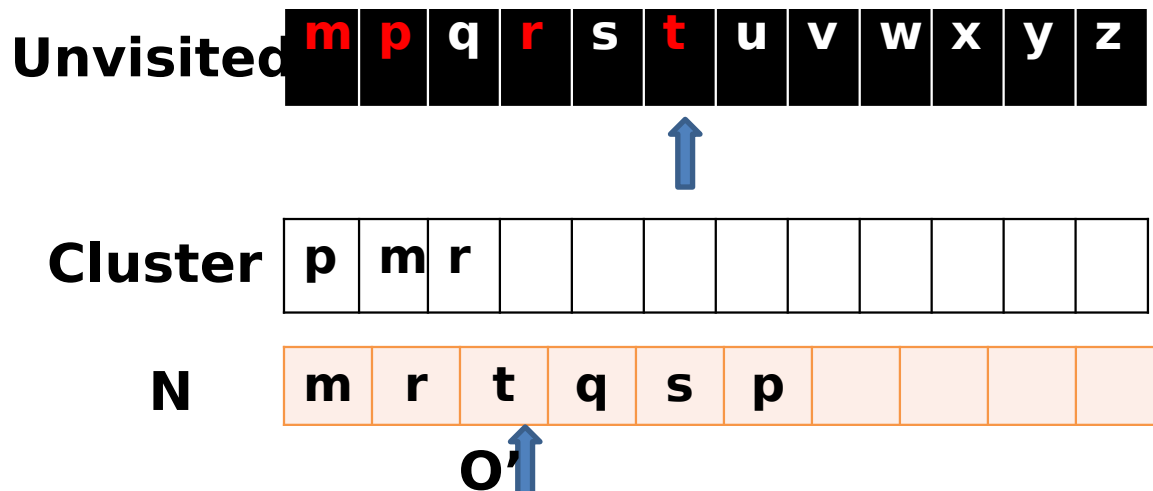
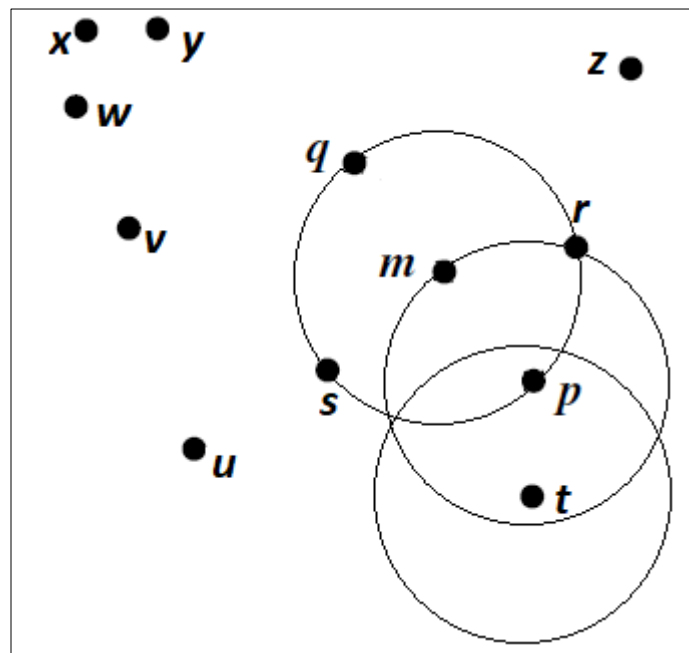
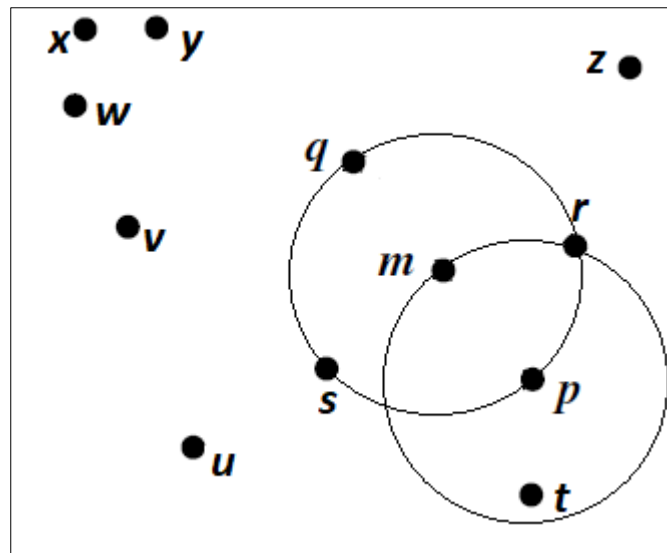
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

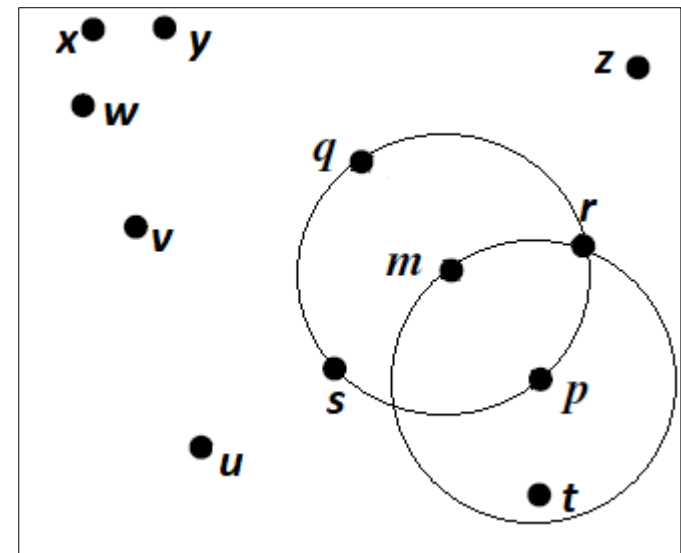
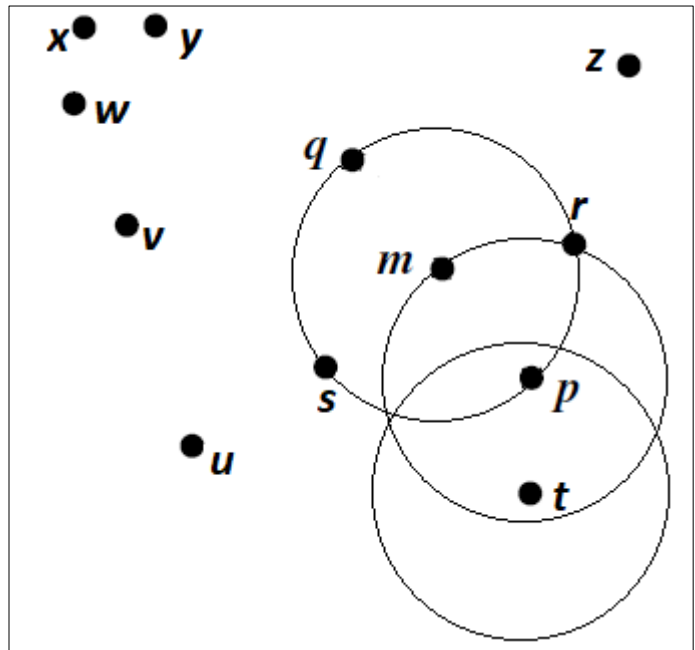


Method:

- ```

(1) mark all objects as unvisited;
(2) do
(3) randomly select an unvisited object O ;
(4) mark O as visited;
(5) if the ϵ -neighborhood of O has at least $MinPts$ objects
(6) create a new cluster C , and add O to C ;
(7) let N be the set of objects in the ϵ -neighborhood of O ;
(8) for each point O' in N
(9) if O' is unvisited
(10) mark O' as visited;
(11) if the ϵ -neighborhood of O' has at least $MinPts$ points,
 add those points to N ;
(12) if O' is not yet a member of any cluster, add O' to C ;
(13) end for
(14) output C ;
(15) else mark O as noise;
(16) until no object is unvisited;

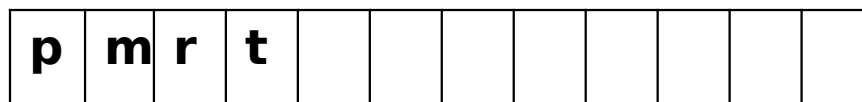
```



Unvisited **m** **p** **q** **r** **s** **t** **u** **v** **w** **x** **y** **z**



## Cluster



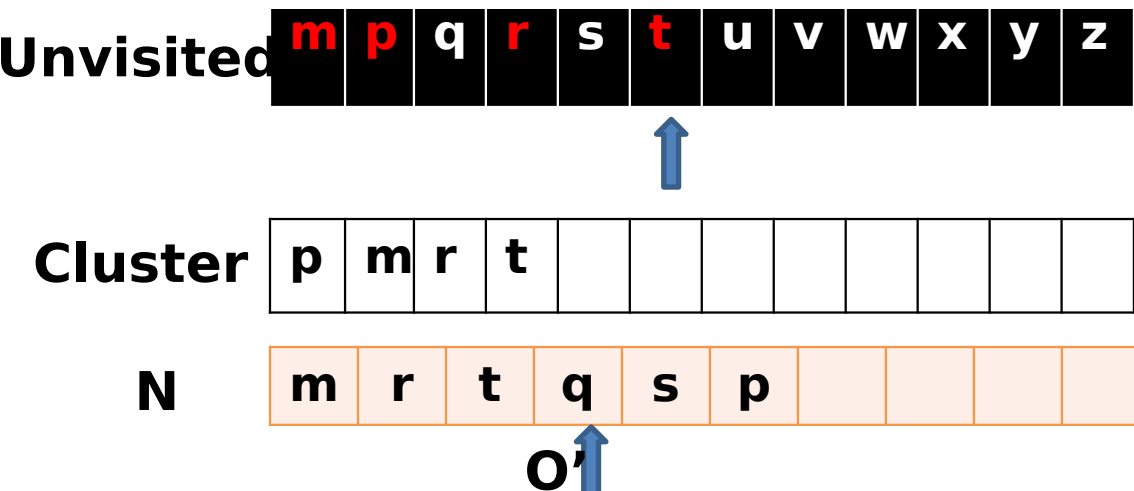
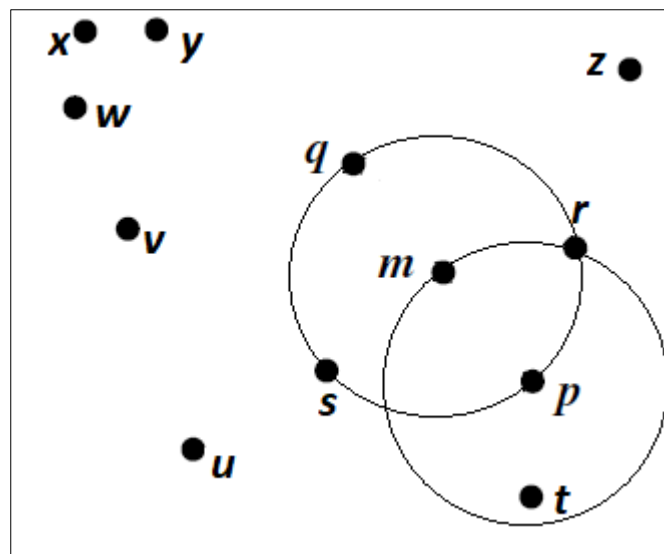
# N



0

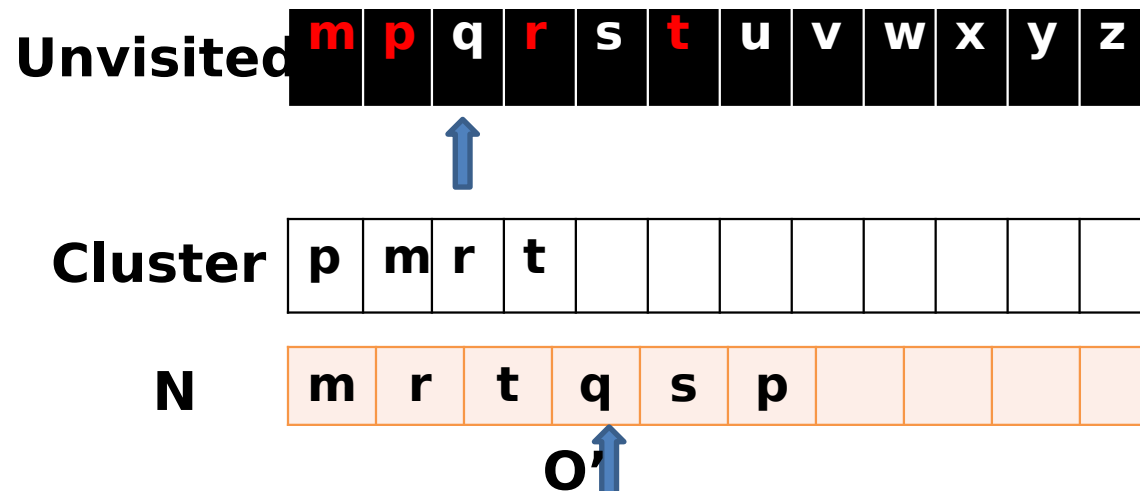
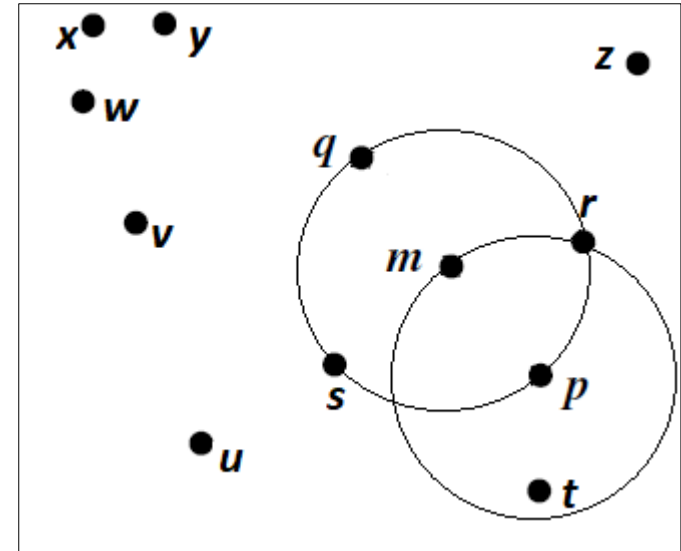
### Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3)     randomly select an unvisited object  $O$ ;
- (4)     mark  $O$  as visited;
- (5)     **if** the  $\epsilon$ -neighborhood of  $O$  has at least MinPts objects
- (6)         create a new cluster  $C$ , and add  $O$  to  $C$ ;
- (7)         let  $N$  be the set of objects in the  $\epsilon$ -neighborhood of  $O$ ;
- (8)         **for** each point  $O'$  in  $N$
- (9)             **if**  $O'$  is unvisited
- (10)                 mark  $O'$  as visited;
- (11)                 **if** the  $\epsilon$ -neighborhood of  $O'$  has at least MinPts points,  
                    add those points to  $N$ ;
- (12)             **if**  $O'$  is not yet a member of any cluster, add  $O'$  to  $C$ ;
- (13)         **end for**
- (14)         output  $C$ ;
- (15)     **else** mark  $O$  as noise;
- (16) **until** no object is unvisited;



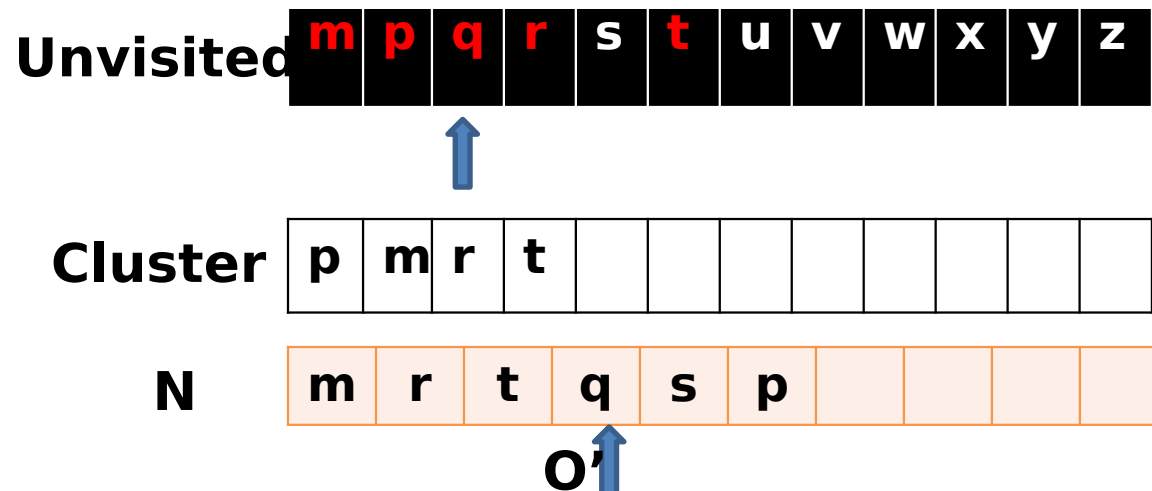
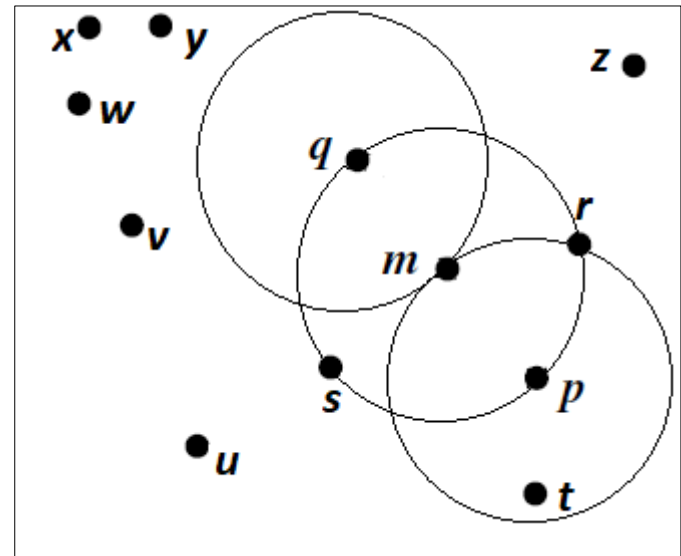
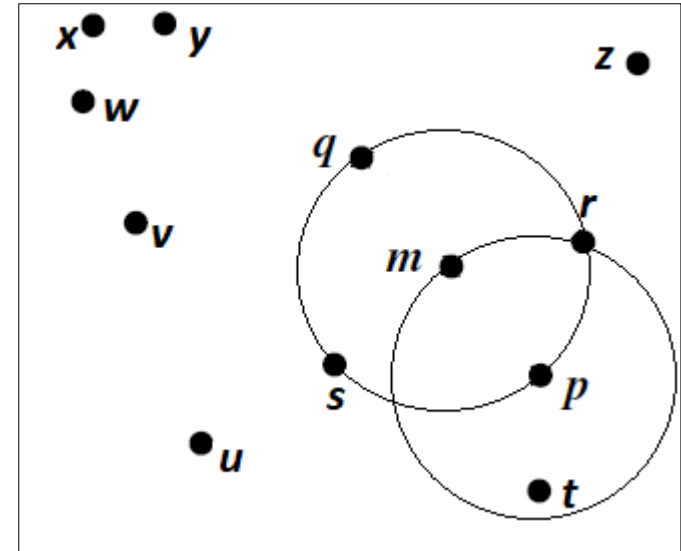
### Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3)     randomly select an unvisited object  $O$ ;
- (4)     mark  $O$  as visited;
- (5)     **if** the  $\epsilon$ -neighborhood of  $O$  has at least MinPts objects
- (6)         create a new cluster  $C$ , and add  $O$  to  $C$ ;
- (7)         let  $N$  be the set of objects in the  $\epsilon$ -neighborhood of  $O$ ;
- (8)         **for** each point  $O'$  in  $N$
- (9)             **if**  $O'$  is unvisited
- (10)                 mark  $O'$  as visited;
- (11)                 **if** the  $\epsilon$ -neighborhood of  $O'$  has at least MinPts points,  
                    add those points to  $N$ ;
- (12)             **if**  $O'$  is not yet a member of any cluster, add  $O'$  to  $C$ ;
- (13)         **end for**
- (14)         output  $C$ ;
- (15)     **else** mark  $O$  as noise;
- (16) **until** no object is unvisited;



### Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3)     randomly select an unvisited object  $O$ ;
- (4)     mark  $O$  as visited;
- (5)     **if** the  $\epsilon$ -neighborhood of  $O$  has at least MinPts objects
- (6)         create a new cluster  $C$ , and add  $O$  to  $C$ ;
- (7)         let  $N$  be the set of objects in the  $\epsilon$ -neighborhood of  $O$ ;
- (8)         **for** each point  $O'$  in  $N$
- (9)             **if**  $O'$  is unvisited
- (10)                 mark  $O'$  as visited;
- (11)                 **if** the  $\epsilon$ -neighborhood of  $O'$  has at least MinPts points,  
                    add those points to  $N$ ;
- (12)             **if**  $O'$  is not yet a member of any cluster, add  $O'$  to  $C$ ;
- (13)         **end for**
- (14)         output  $C$ ;
- (15)     **else** mark  $O$  as noise;
- (16) **until** no object is unvisited;

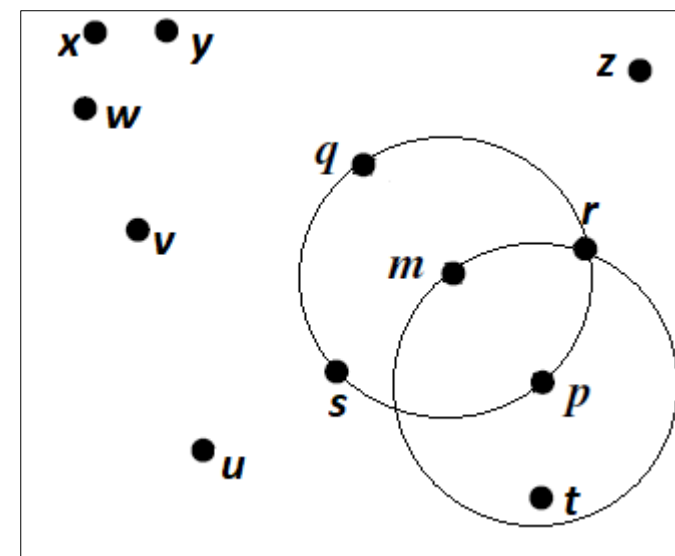
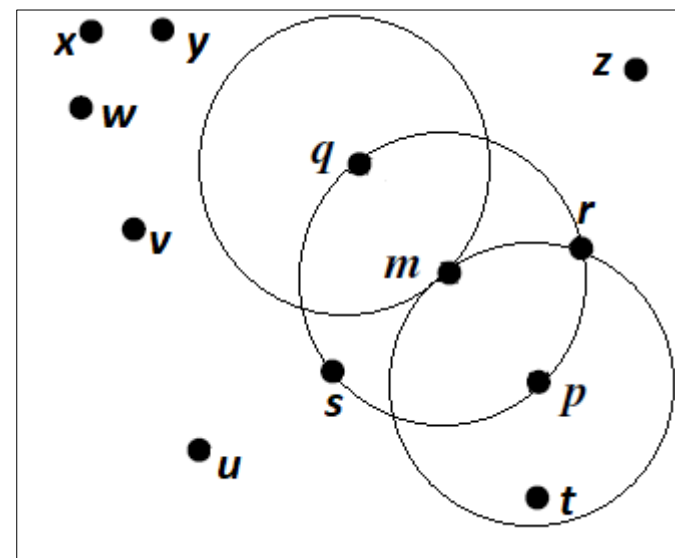


### Method:

- ```

(1) mark all objects as unvisited;
(2) do
(3)     randomly select an unvisited object  $O$ ;
(4)     mark  $O$  as visited;
(5)     if the  $\epsilon$ -neighborhood of  $O$  has at least  $MinPts$  objects
(6)         create a new cluster  $C$ , and add  $O$  to  $C$ ;
(7)         let  $N$  be the set of objects in the  $\epsilon$ -neighborhood of  $O$ ;
(8)         for each point  $O'$  in  $N$ 
(9)             if  $O'$  is unvisited
(10)                mark  $O'$  as visited;
(11)                if the  $\epsilon$ -neighborhood of  $O'$  has at least  $MinPts$  points,
                    add those points to  $N$ ;
(12)                if  $O'$  is not yet a member of any cluster, add  $O'$  to  $C$ ;
(13)         end for
(14)     output  $C$ ;
(15) else mark  $O$  as noise;
(16) until no object is unvisited;

```



Unvisited **m p q r s t u v w x y z**



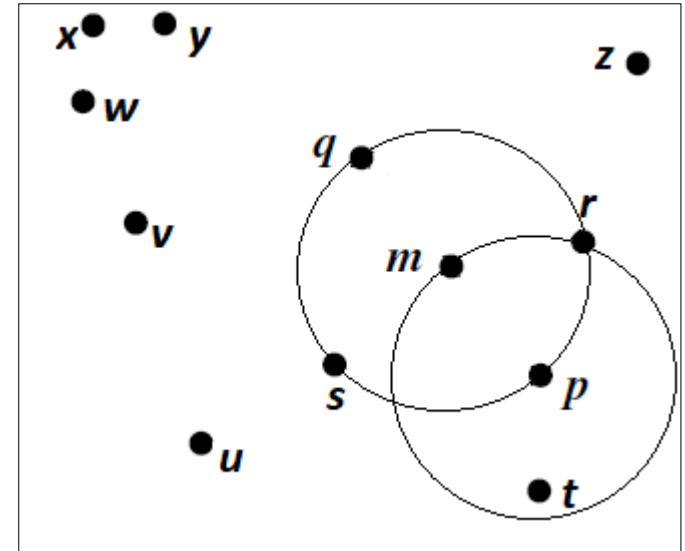
Cluster	p	m	r	t	q						
---------	---	---	---	---	---	--	--	--	--	--	--

N	m	r	t	q	s	p				
---	---	---	---	---	---	---	--	--	--	--



Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---



Cluster

p	m	r	t	q							
---	---	---	---	---	--	--	--	--	--	--	--

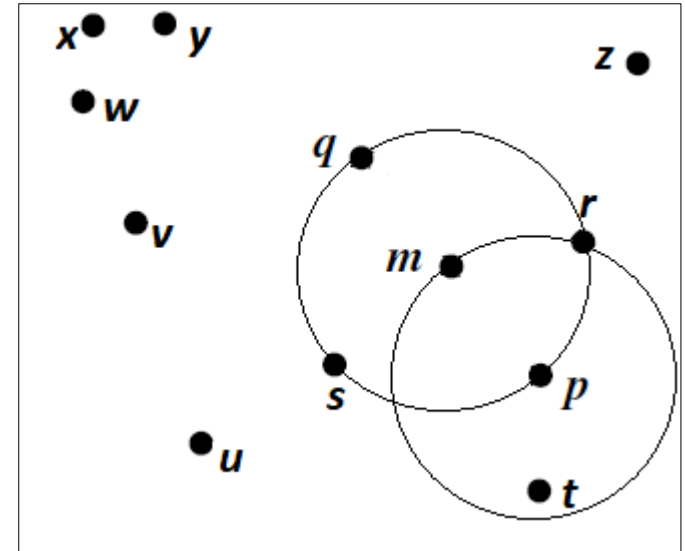
N

m	r	t	q	s	p						
---	---	---	---	---	---	--	--	--	--	--	--

O'

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

Cluster

p	m	r	t	q							
---	---	---	---	---	--	--	--	--	--	--	--

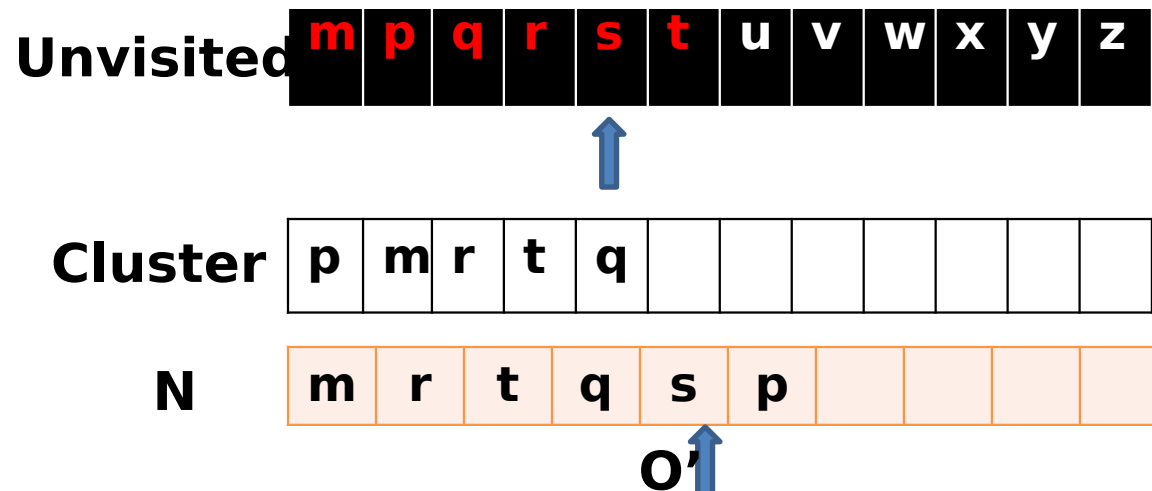
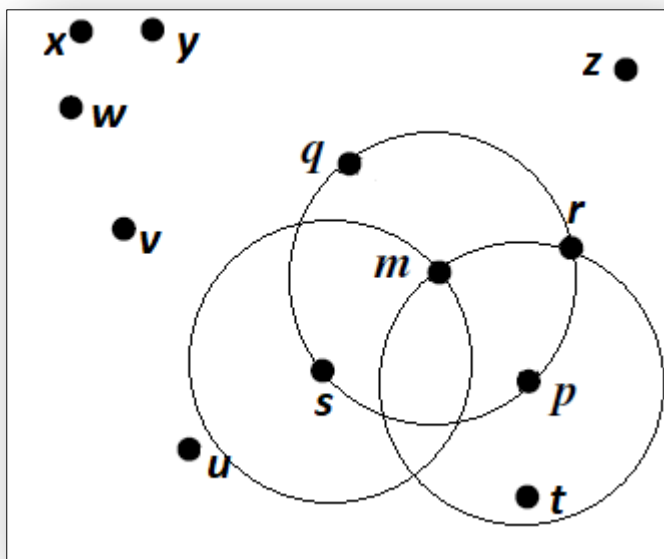
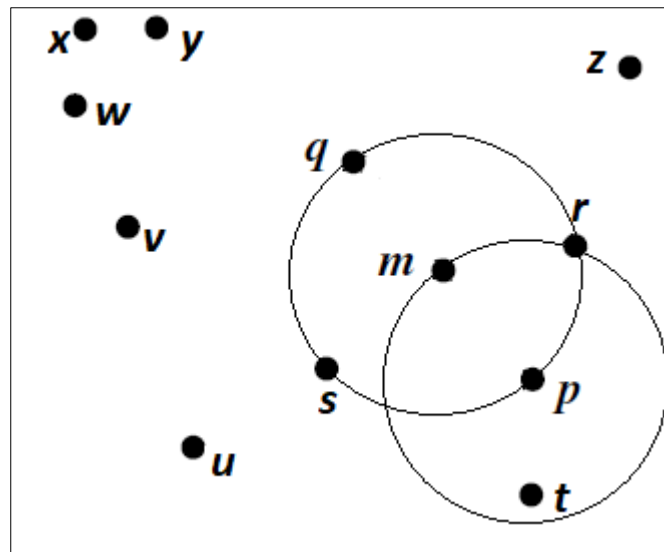
N

m	r	t	q	s	p						
---	---	---	---	---	---	--	--	--	--	--	--

O'

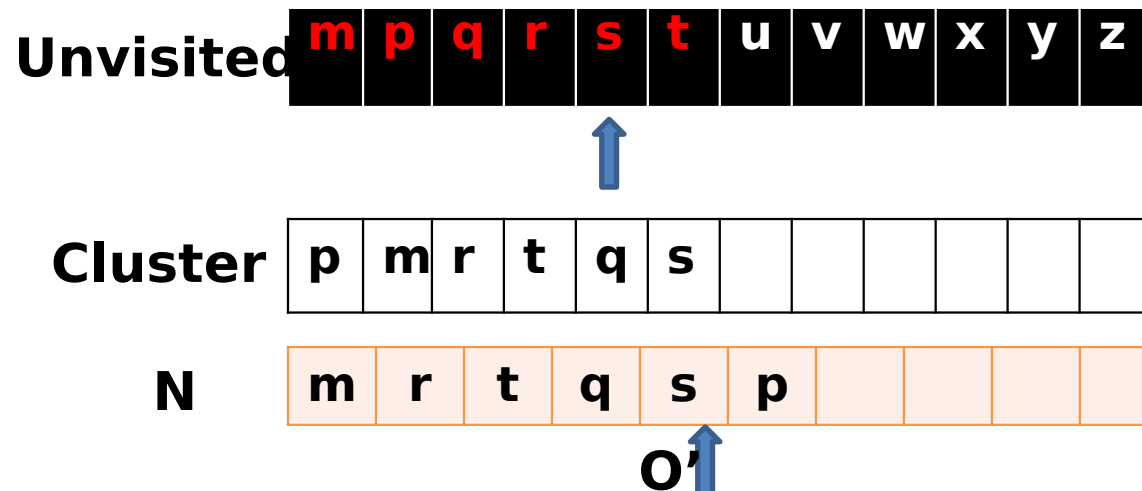
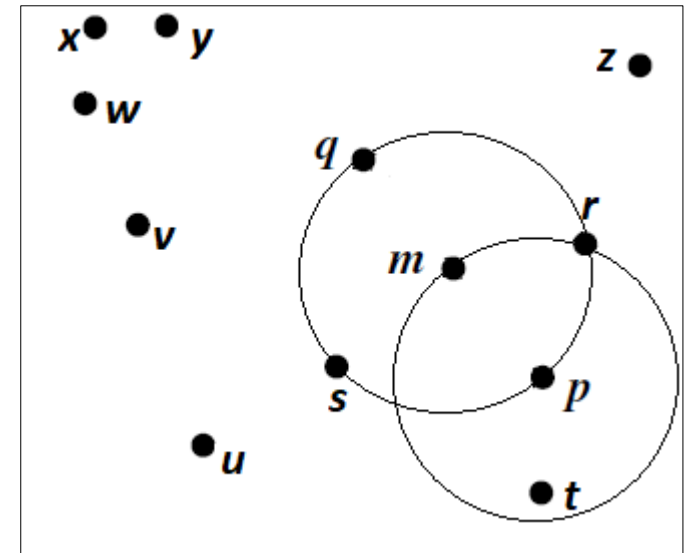
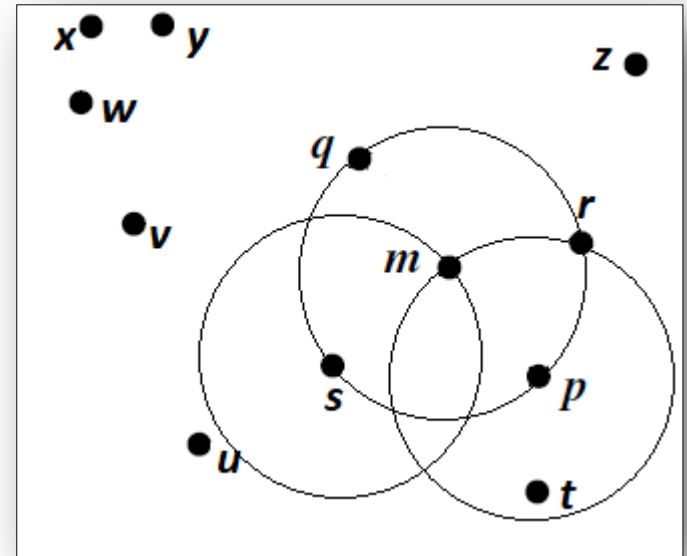
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



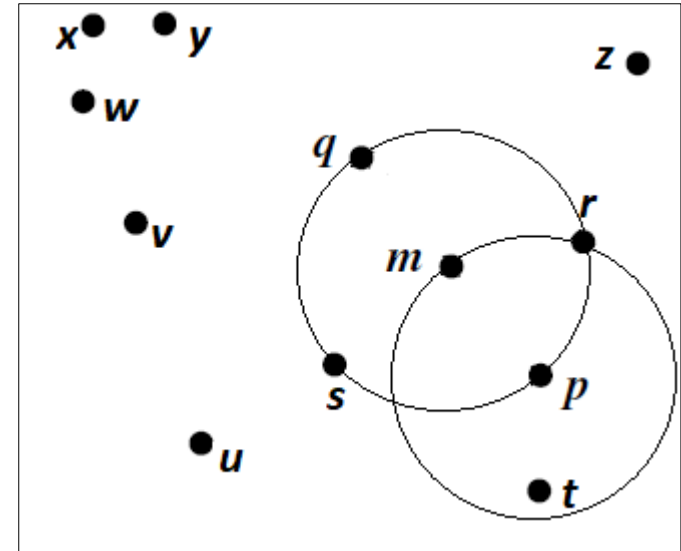
Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---



Cluster

p	m	r	t	q	s						
---	---	---	---	---	---	--	--	--	--	--	--

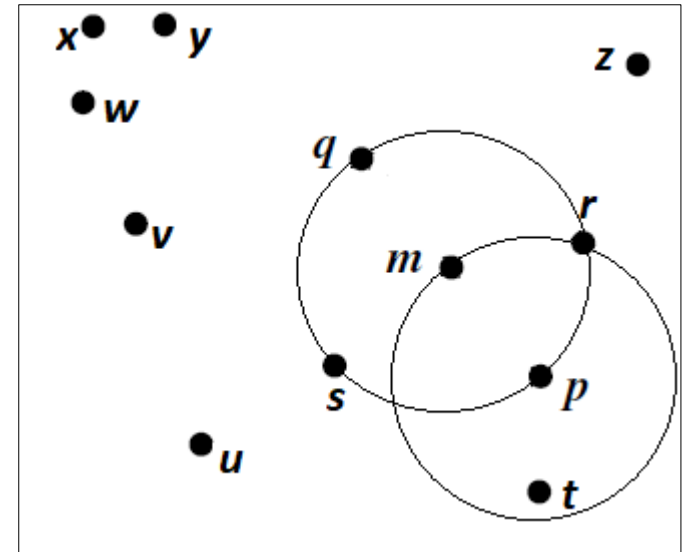
N

m	r	t	q	s	p						
---	---	---	---	---	---	--	--	--	--	--	--

O'

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited **m p q r s t u v w x y z**



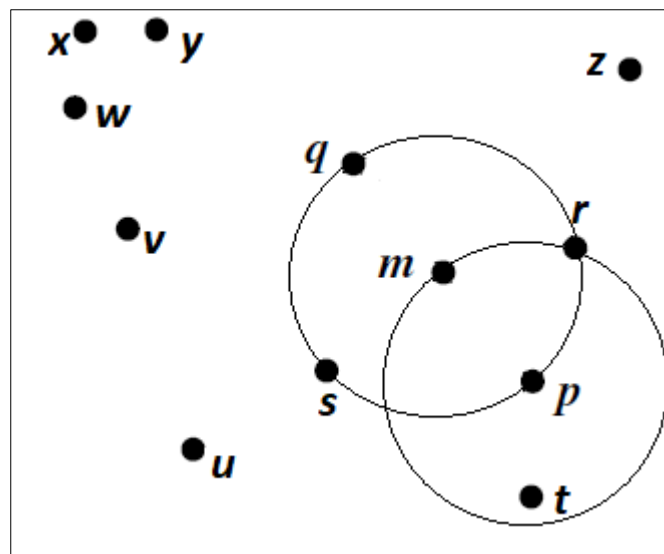
Cluster **p m r t q s**

N **m r t q s p**



Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited **m p q r s t** u v w x y z

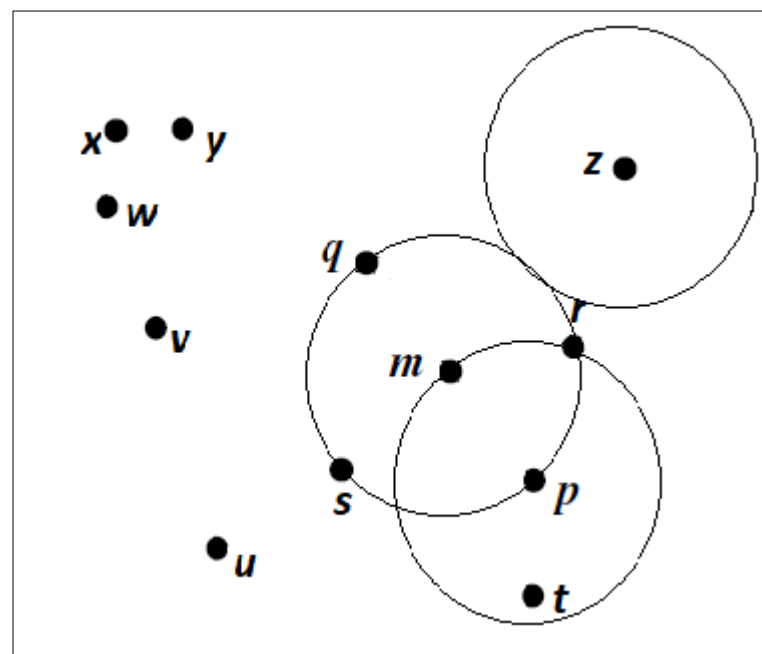
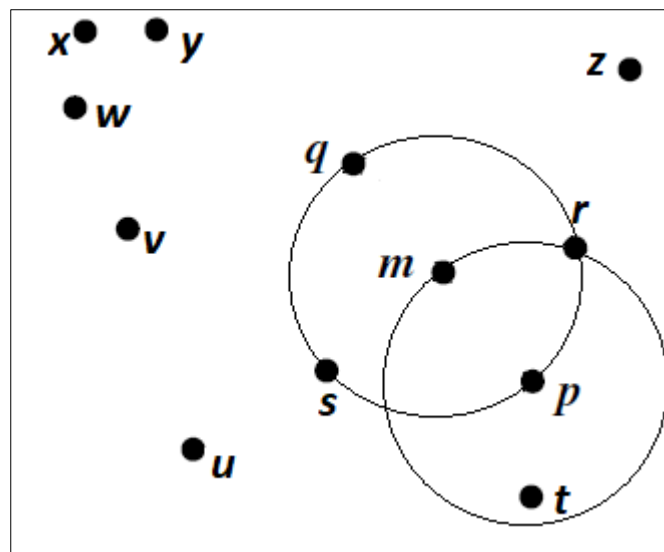
↑

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

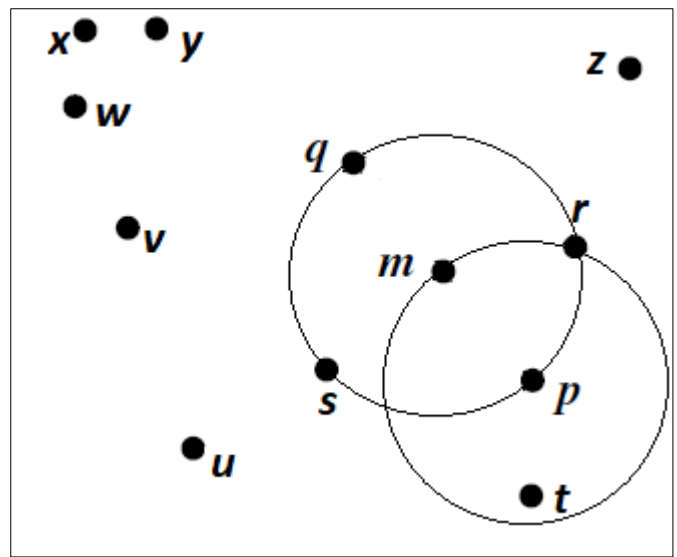


Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

↑

Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
- add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

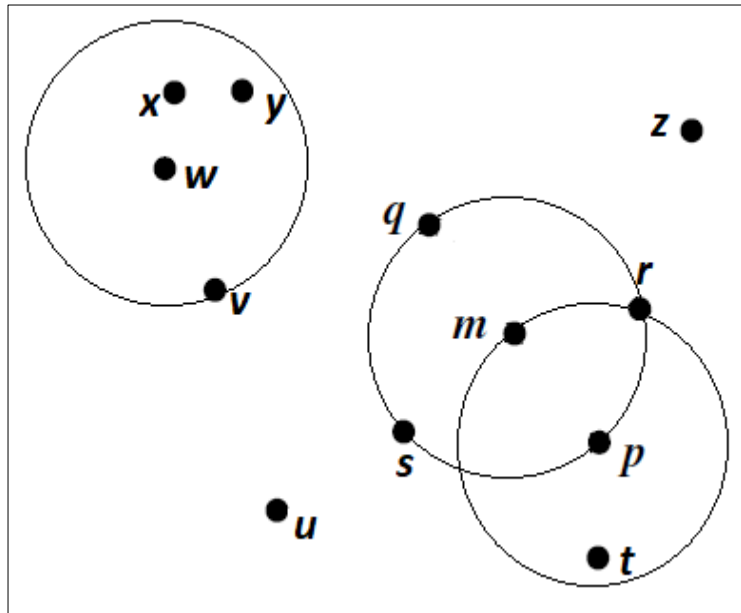
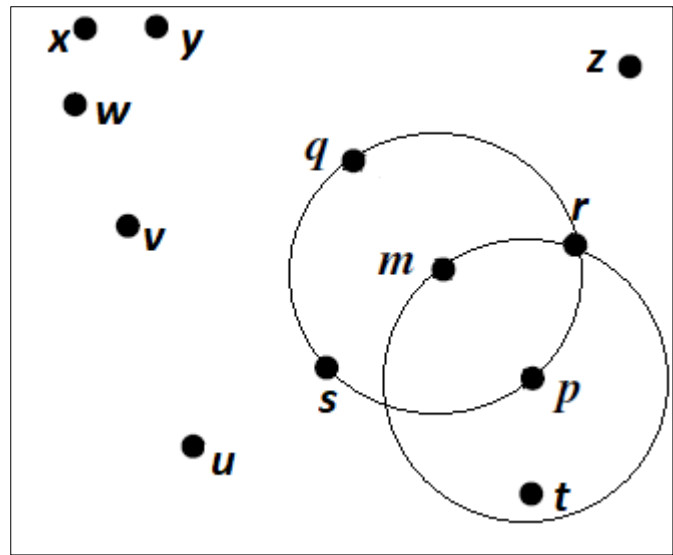
Cluster

w											
---	--	--	--	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

O'



Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

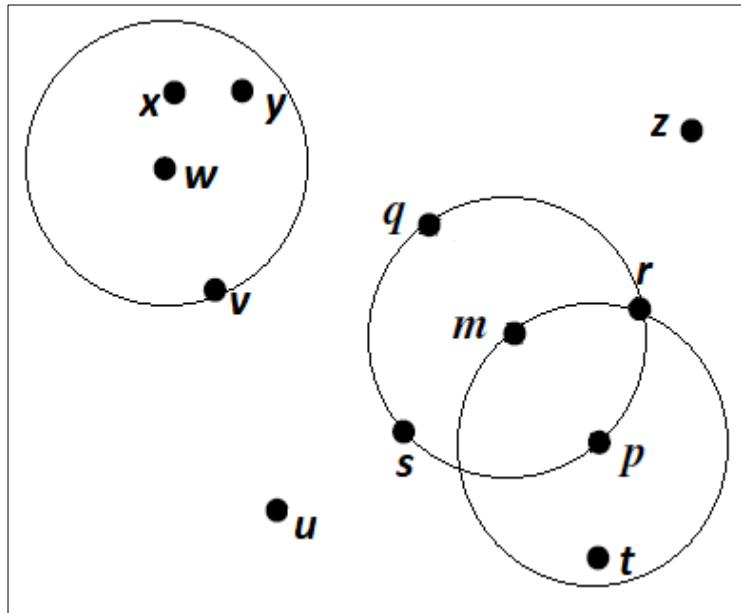
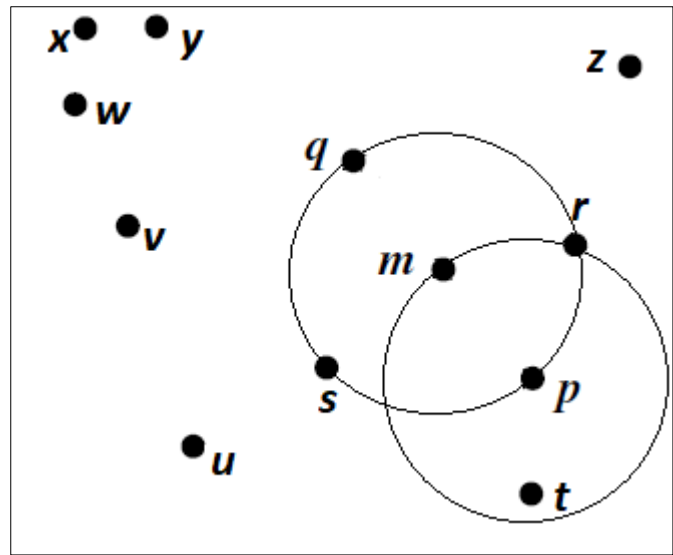
Cluster

w											
---	--	--	--	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

O'



Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

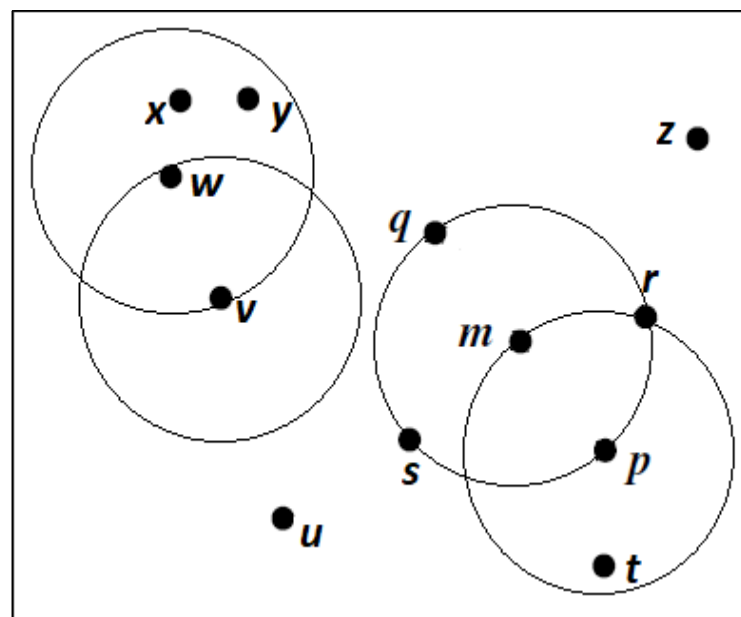
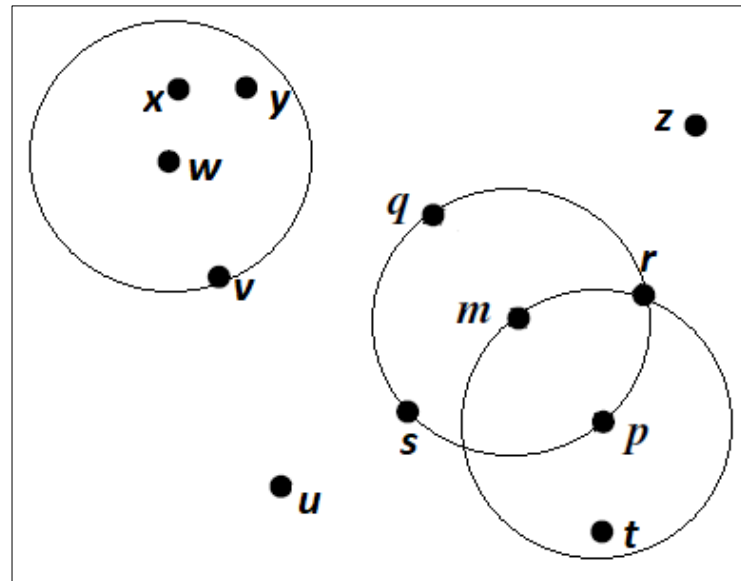
Cluster

w											
---	--	--	--	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

O' 



Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

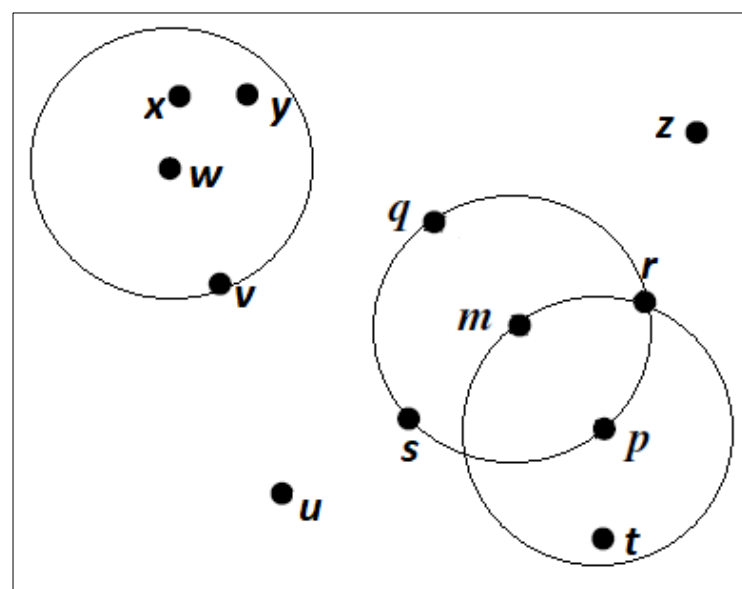
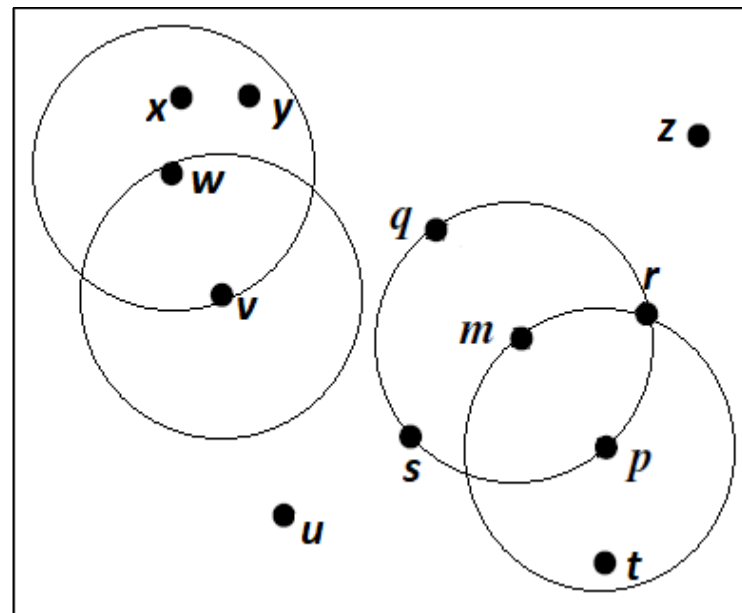
Cluster

w	v										
---	---	--	--	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

O' 

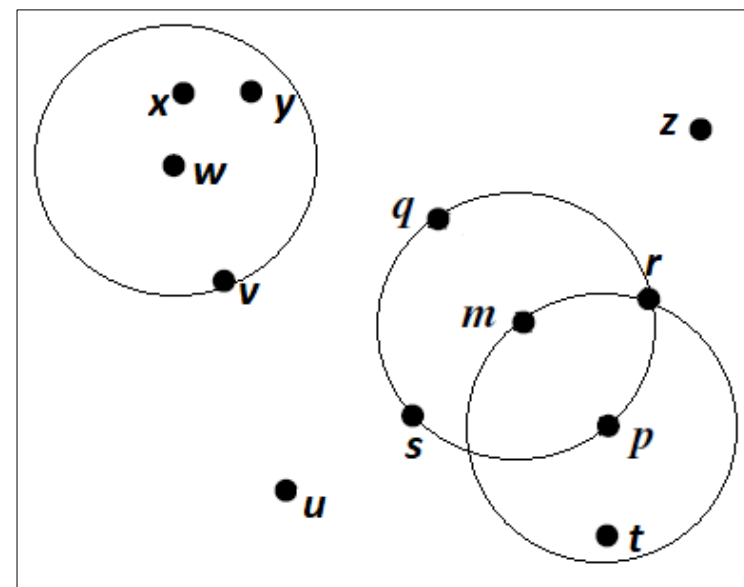


Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited **m p q r s t u v w x y z**



Cluster

w	v										
---	---	--	--	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

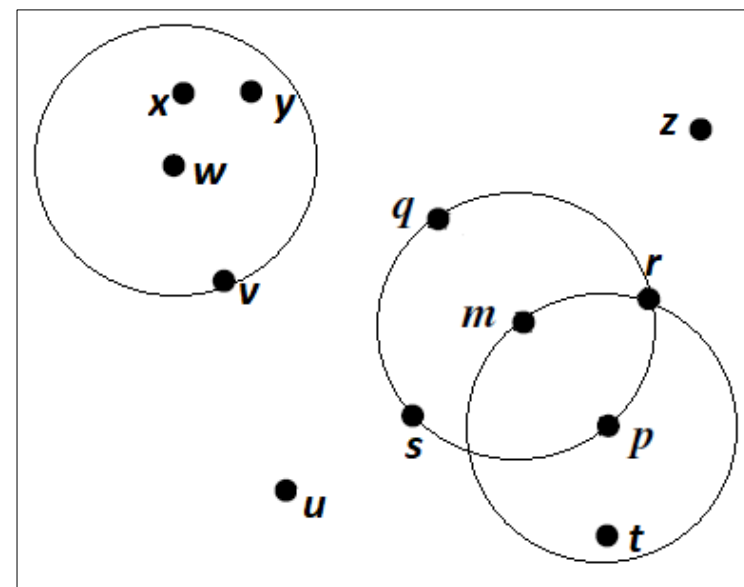
O'

Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited **m p q r s t u v w x y z**

Cluster

w	v										
---	---	--	--	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

O'

Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

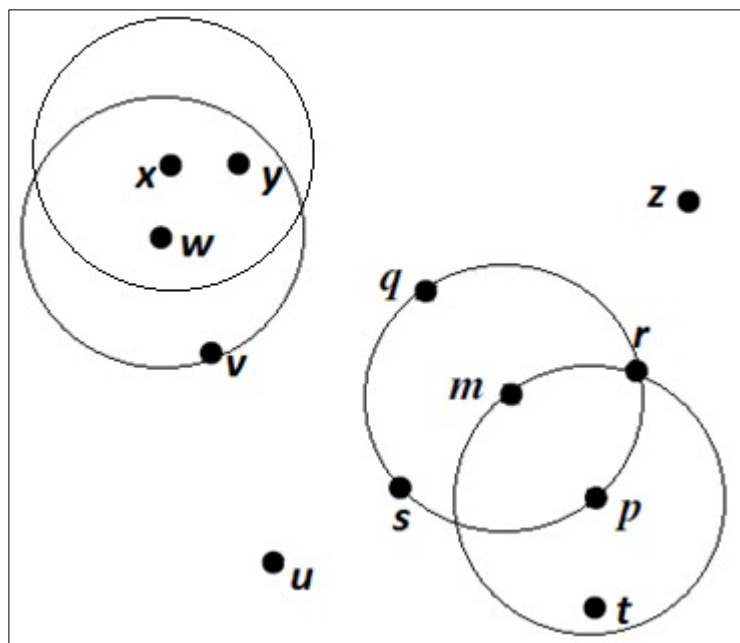
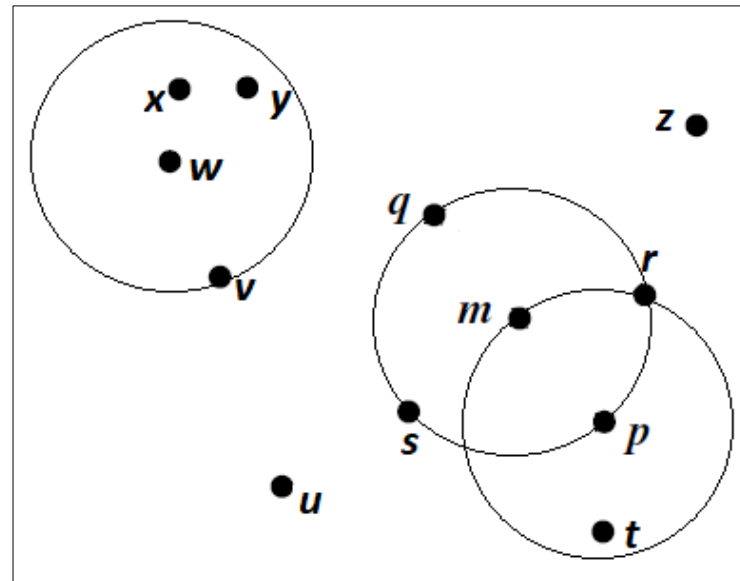
Cluster

w	v										
---	---	--	--	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

O'



Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

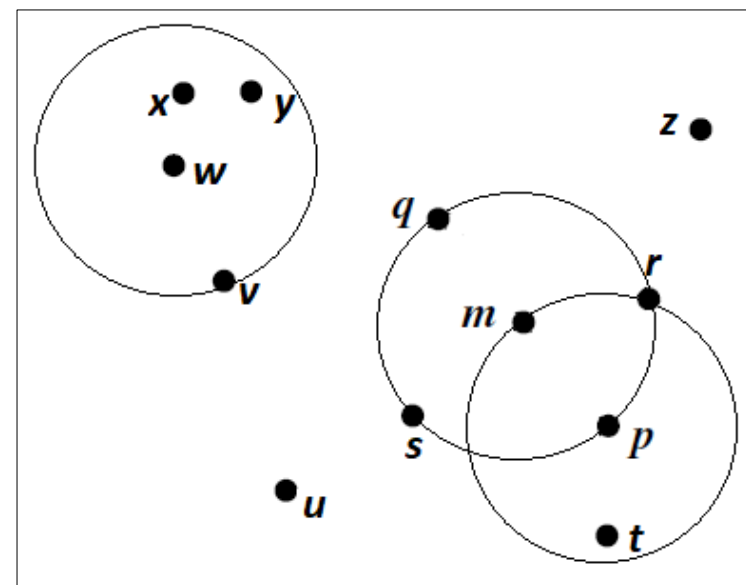
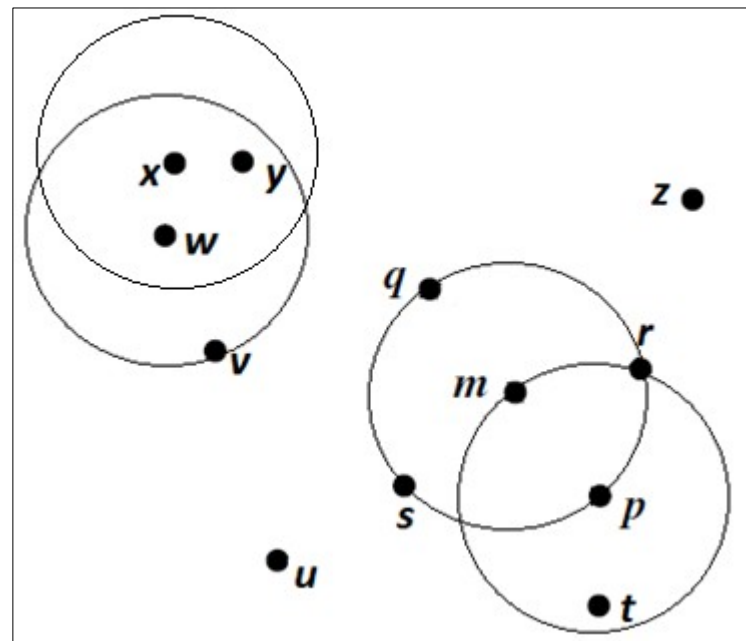
Cluster

w	v	x									
---	---	---	--	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

O'



Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

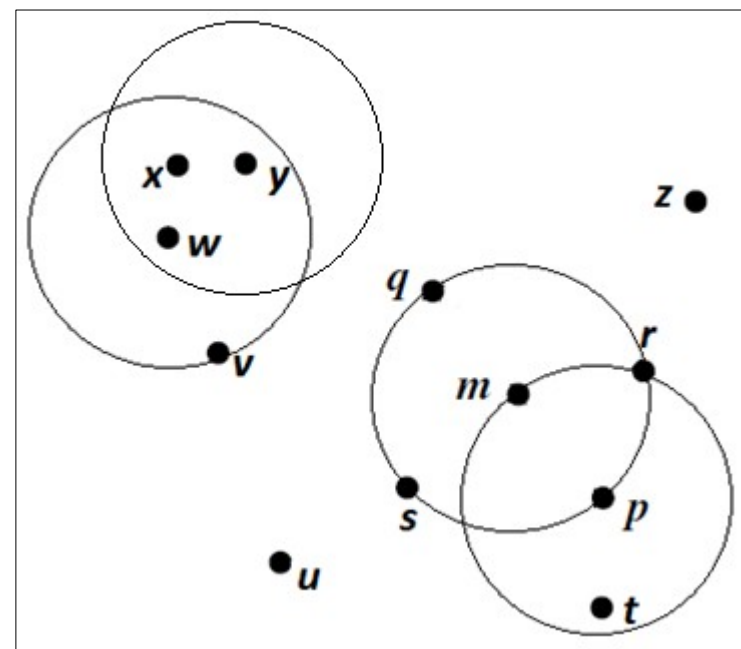
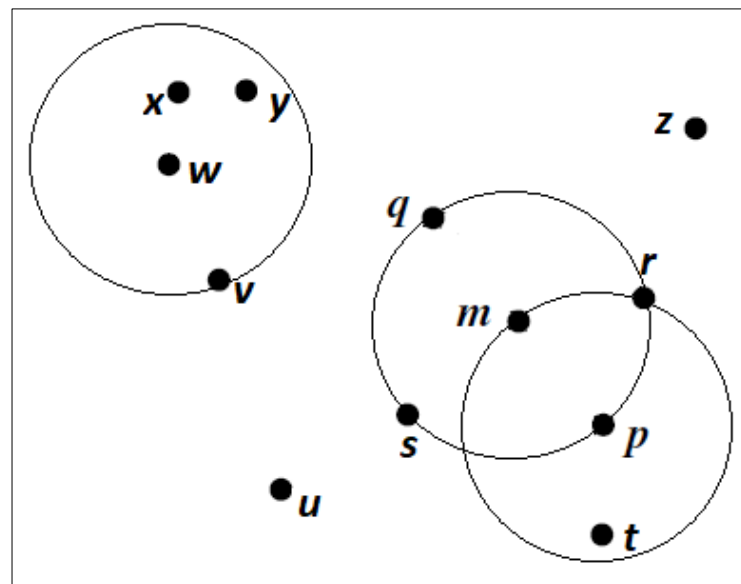
Cluster

w	v	x									
---	---	---	--	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

O'

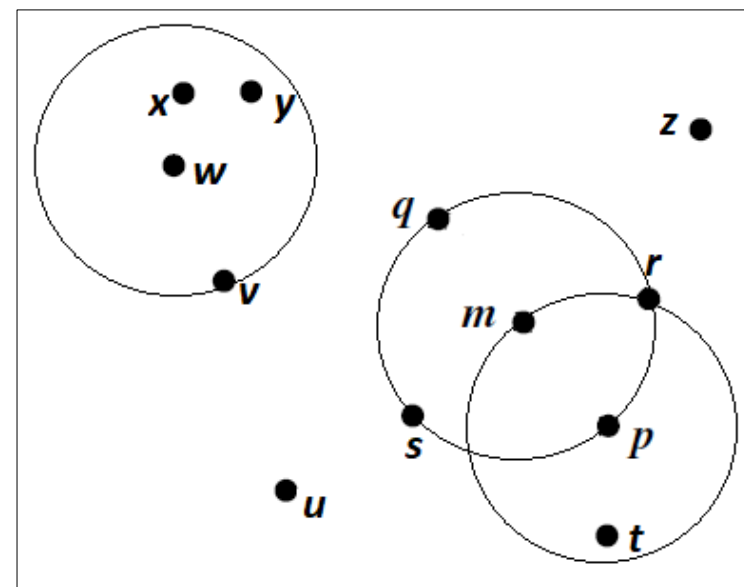


Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited **m p q r s t u v w x y z**

Cluster

w	v	x									
---	---	---	--	--	--	--	--	--	--	--	--

N

v	x	y								
---	---	---	--	--	--	--	--	--	--	--

O' ↑

Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

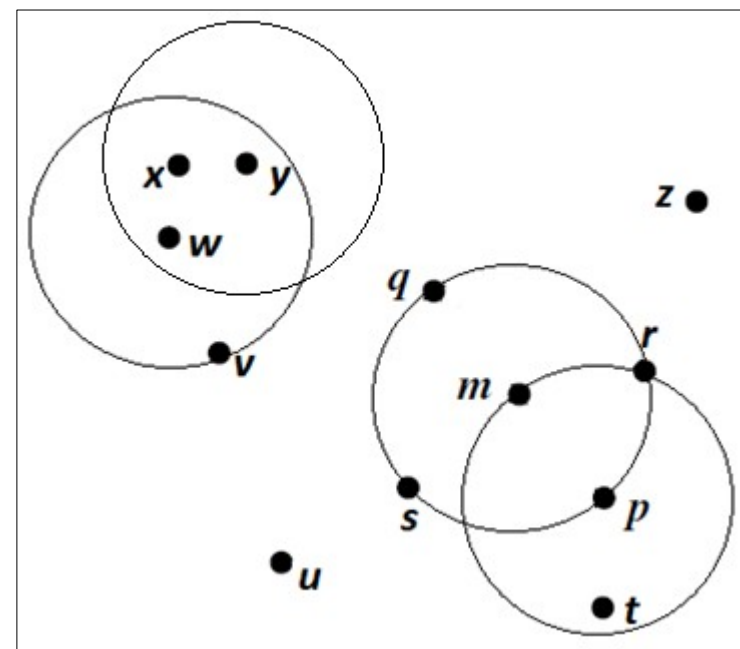
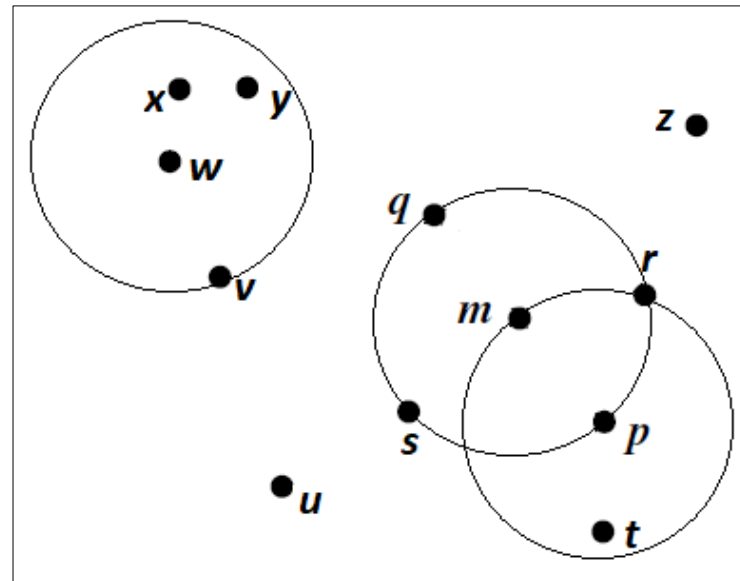
Cluster

w	v	x									
---	---	---	--	--	--	--	--	--	--	--	--

N

v	x	y								
---	---	---	--	--	--	--	--	--	--	--

O'



Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

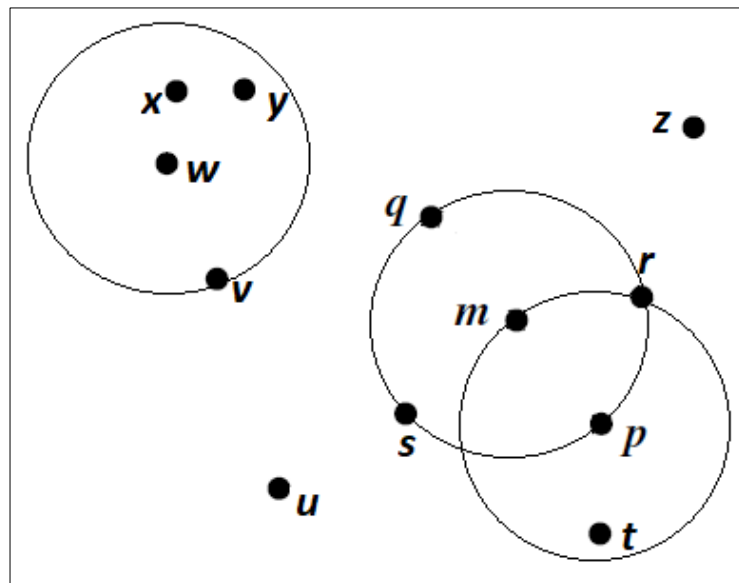
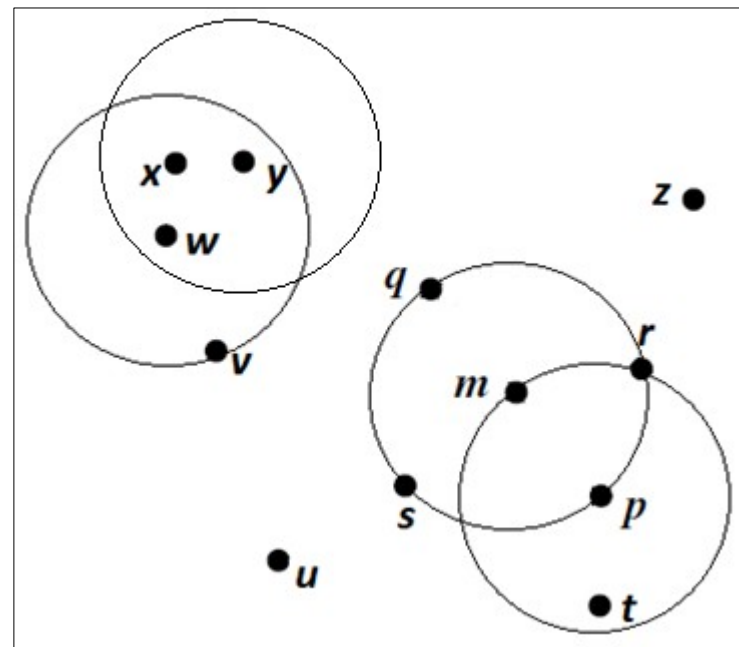
Cluster

w	v	x	y								
---	---	---	---	--	--	--	--	--	--	--	--

N

v	x	y									
---	---	---	--	--	--	--	--	--	--	--	--

 ↑
 O'

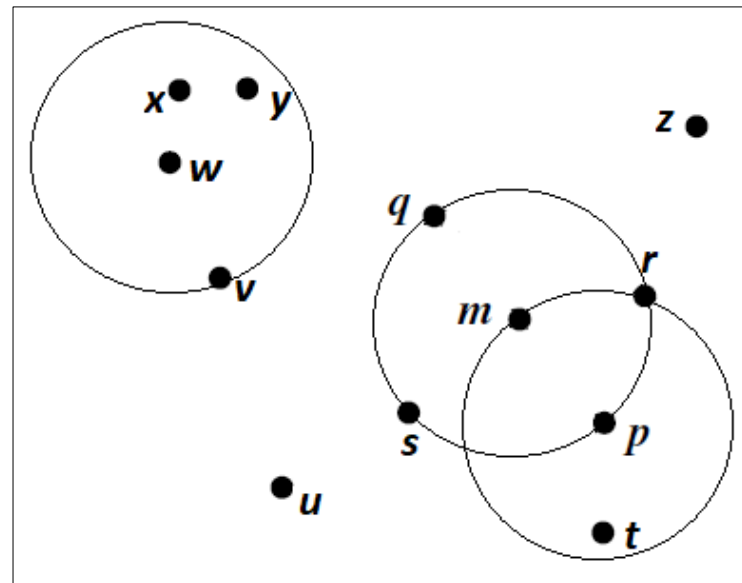


Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited

m	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---

↑

Noise

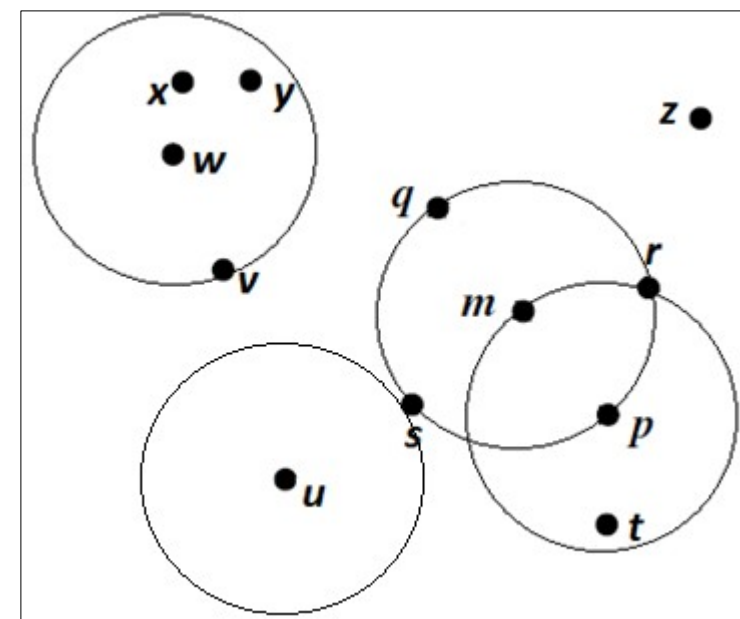
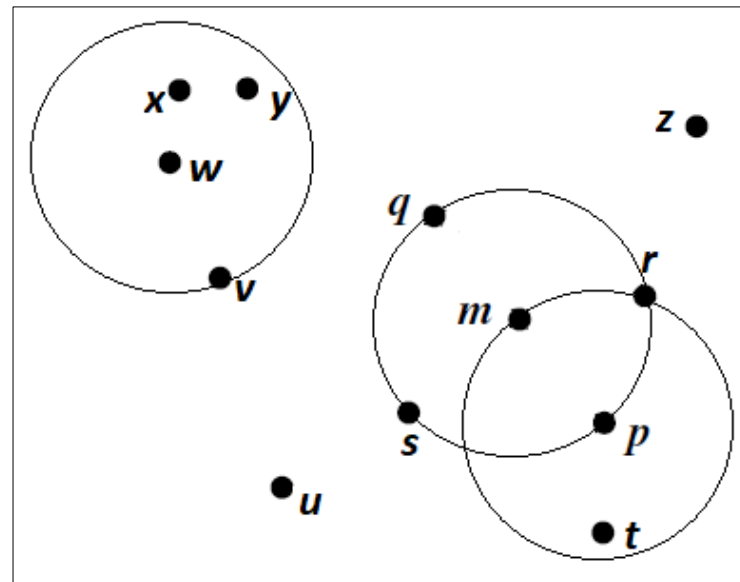
z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;

Unvisited **m p q r s t u v w x y z**

↑

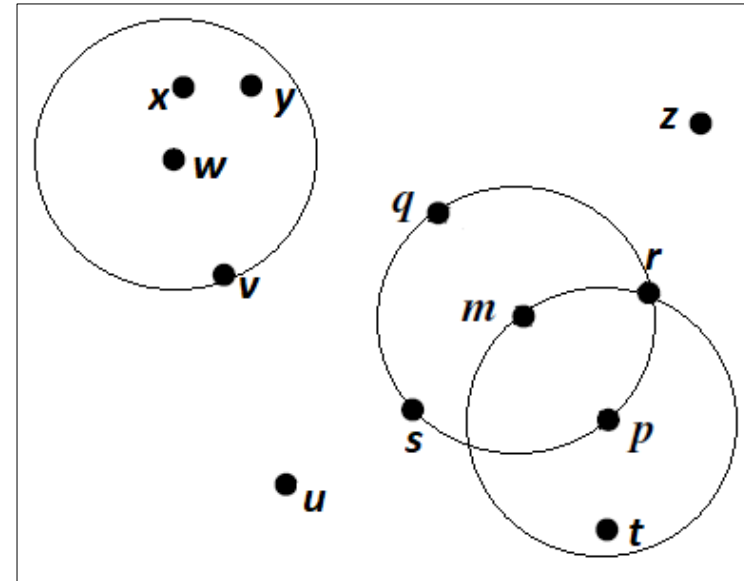


Noise

z				
---	--	--	--	--

Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object O ;
- (4) mark O as visited;
- (5) **if** the ϵ -neighborhood of O has at least MinPts objects
- (6) create a new cluster C , and add O to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of O ;
- (8) **for** each point O' in N
- (9) **if** O' is unvisited
- (10) mark O' as visited;
- (11) **if** the ϵ -neighborhood of O' has at least MinPts points,
 add those points to N ;
- (12) **if** O' is not yet a member of any cluster, add O' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark O as noise;
- (16) **until** no object is unvisited;



Unvisited **m p q r s t u v w x y z**



Cluster

p	m	r	t	q	s						
---	---	---	---	---	---	--	--	--	--	--	--

Cluster

w	v	x	y								
---	---	---	---	--	--	--	--	--	--	--	--

Noise

z	u			
---	---	--	--	--

Thank You