WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Data Clustering

Marzena Kryszkiewicz

HUMAN CAPITAL
Project is co-financed by European Union within European Social Fund

1

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

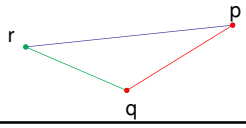
Basic Notions

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Distance Metric

A *distance metric* is defined as a function that satisfies the following conditions:

- ∀p,q, distance(p, q) ≥ 0;
- ∀p, distance(p, p) = 0;
- ∀p,q, distance(p, q) = distance(q, p);
- ∀p,q,r, distance(p, r) ≤ distance(p, q) + distance(q, r) /* triangle inequality property */.



3

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Example Distance Metrics

- Euclidean(p,q) = $\sqrt{\sum_{i=1..n}(p_i - q_i)^2}$
- Manhattan(p,q) = $\sum_{i=1..n}|p_i - q_i|$
- Minkowski(p,q) = $\sqrt[m]{\sum_{i=1..n}|p_i - q_i|^m}$

(Minkowski function is a distance metric for $m \geq 1$).

4

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

DBSCAN: Density-Based Clustering with Noise

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Eps-Neighborhood

- *Eps-neighborhood of a point p* (denoted by $N_{Eps}(p)$) is defined as the set of all points q in dataset D that are distant from p by no more than Eps; that is,

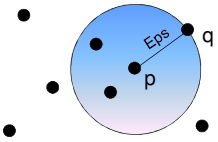
$N_{Eps}(p) = \{q \in D \mid distance(p,q) \leq Eps\}$.

6

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Eps-Neighborhood

$|N_{Eps}(p)| = 4$.



7

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Core Points

- A point p is defined as a *core point* if its *Eps-neighborhood* contains at least MinPts points; that is, if $|N_{Eps}(p)| \geq$ MinPts.

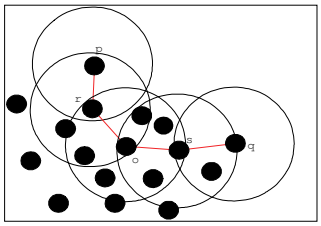
8

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Core Points

For MinPts = 6:

- r is a core point;
- p is not a core.



9

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Clusters and Noise in DBSCAN

- A core point is treated as a seed from which a cluster is built.
- Whenever any core point is included in the cluster, all points in its Eps-neighborhood are also included in the cluster unless they were earlier assigned to another cluster.
- A point that is not included in any cluster (that is, does not belong to Eps-neighborhood of any core point) is called a *noise point*.

10

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

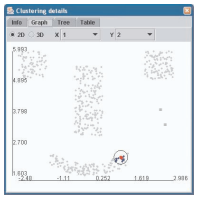
Non-Core Points in DBSCAN

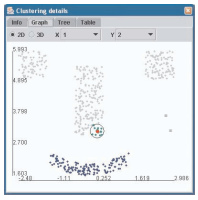
- **Property:** A noise point is a non-core point that does not belong to any cluster (that is, which does not belong to Eps-neighborhood of any core point).
- A *border point* is a non-core point that belongs to a cluster (that is, which belongs to Eps-neighbourhood of at least one core point).

11

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Clusters in DBSCAN…

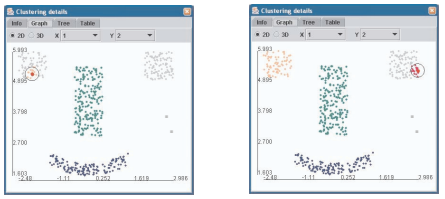




12

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

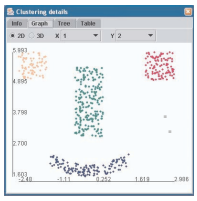
Clusters in DBSCAN…



13

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

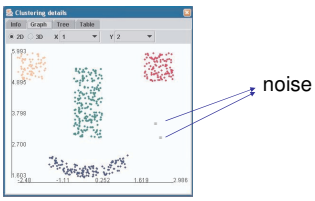
Clusters in DBSCAN



14

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

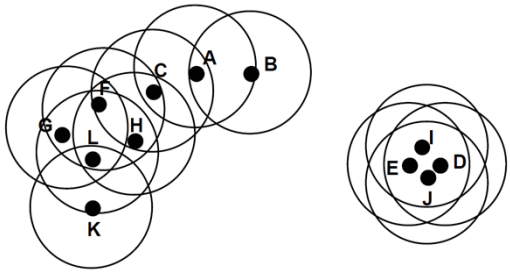
Clusters and Noise in DBSCAN



15

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

DBSCAN – Definition's Illustration



minPts = 4

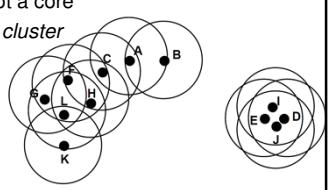
16

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

DBSCAN – Extension of a Cluster

- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing



minPts = 4

17

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

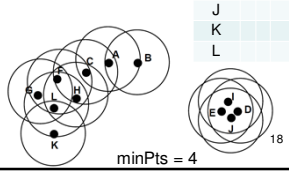
DBSCAN – Example Execution…

- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 1.
- $N_{Eps}(A)$ = {A, B, C} – A is non-core.

q∈D	x	y	ClId
A			N
B			
C			
D			
E			
F			
G			
H			
I			
J			
K			
L			



minPts = 4

18

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

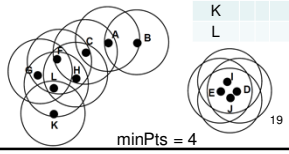
DBSCAN – Example Execution…

- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 1.
- $N_{Eps}(B) = \{B, A\}$ – B is non-core.

q∈D	x	y	ClId
A			N
B			N
C			
D			
E			
F			
G			
H			
I			
J			
K			
L			



minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

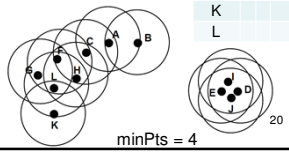
DBSCAN – Example Execution…

- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 1.
- $N_{Eps}(C) = \{C, A, F, H\}$ – C is a core.
- seeds = {A, F, H}.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			
E			
F			1
G			
H			1
I			
J			
K			
L			



minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

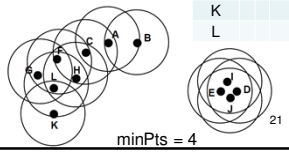
DBSCAN – Example Execution…

- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 1.
- seeds = {A, F, H}.
- $N_{Eps}(A) = \{A, B, C\}$ – A is non-core.
- Updated seeds = {F, H}.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			
E			
F			1
G			
H			1
I			
J			
K			
L			



minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

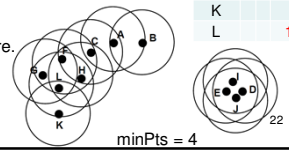
DBSCAN – Example Execution…

- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 1.
- seeds = {F, H}.
- $N_{Eps}(F) = \{F, C, G, H, L\}$ – F is a core.
- Updated seeds = {H, G, L}.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			
E			
F			1
G			1
H			1
I			
J			
K			
L			1



minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

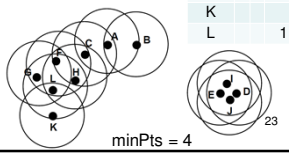
DBSCAN – Example Execution…

- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 1.
- seeds = {H, G, L}.
- $N_{Eps}(H) = \{H, C, F, L\}$ – H is a core.
- Updated seeds = {G, L}.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			
E			
F			1
G			1
H			1
I			
J			
K			
L			1



minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

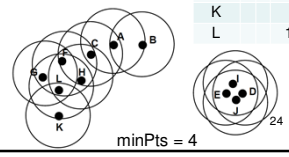
DBSCAN – Example Execution…

- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 1.
- seeds = {G, L}.
- $N_{Eps}(G) = \{G, F, L\}$ – G is non-core.
- Updated seeds = {L}.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			
E			
F			1
G			1
H			1
I			
J			
K			
L			1



minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

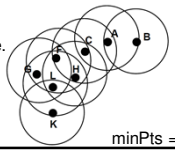
DBSCAN – Example Execution…

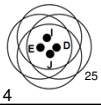
- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 1.
- seeds = {L}.
- N_{Eps}(L) = {L, F, G, H, K} – L is a core.
- Updated seeds = {K}.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			
E			
F			1
G			1
H			1
I			
J			
K			1
L			1





minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

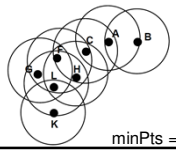
DBSCAN – Example Execution…

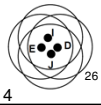
- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 1.
- seeds = {K}.
- N_{Eps}(K) = {K, L} – K is non-core.
- Updated seeds = {}.
- Cluster 1 was created.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			
E			
F			1
G			1
H			1
I			
J			
K			1
L			1





minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

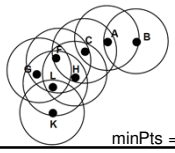
DBSCAN – Example Execution…

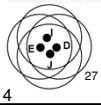
- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 2.
- N_{Eps}(D) = {D, E, I, J} – D is a core.
- seeds = {E, I, J}.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			2
E			2
F			1
G			1
H			1
I			2
J			2
K			1
L			1





minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

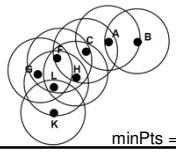
DBSCAN – Example Execution…

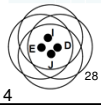
- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 2.
- seeds = {E, I, J}.
- N_{Eps}(E) = {E, D, I, J} – E is a core.
- Updated seeds = {I, J}.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			2
E			2
F			1
G			1
H			1
I			2
J			2
K			1
L			1





minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

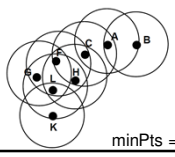
DBSCAN – Example Execution…

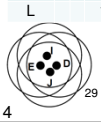
- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 2.
- seeds = {I, J}.
- N_{Eps}(I) = {I, D, E, J} – I is a core.
- Updated seeds = {J}.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			2
E			2
F			1
G			1
H			1
I			2
J			2
K			1
L			1





minPts = 4

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

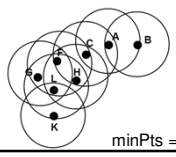
DBSCAN – Example Execution…

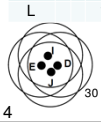
- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

- ClId = 2.
- seeds = {J}.
- N_{Eps}(J) = {J, D, E, I} – J is a core.
- Updated seeds = {}.
- Cluster 2 was created.

q∈D	x	y	ClId
A			~~N~~ 1
B			N
C			1
D			2
E			2
F			1
G			1
H			1
I			2
J			2
K			1
L			1





minPts = 4

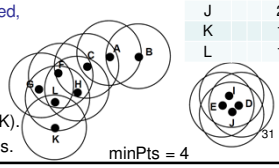
DBSCAN – Example Execution

- Dealing with a point q from an analysed neighborhood:

if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing

q∈D	x	y	Cld
A			~~N~~ 1
B			N
C			1
D			2
E			2
F			1
G			1
H			1
I			2
J			2
K			1
L			1

- All points in dataset D were processed, so the clustering is done.
- The obtained clustering result: 2 clusters and 1 noise point (here: point B).
- Cluster 1 has 3 border points (A, G, K).
- Cluster 2 does not have border points.



minPts = 4

31

Properties of Points in DBSCAN

- For given values of Eps and minPts:
  - DBSCAN determines noise points in a deterministic way independently of the order of processing points in dataset D.
  - DBSCAN assigns each core point exactly to one cluster in a deterministic way independently of the order of processing points in dataset D.
  - DBSCAN assigns a border point to one cluster. The cluster to which it will be assigned depends on the order of processing points in dataset D.

32

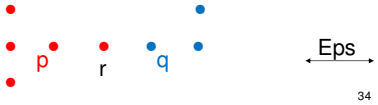
Illustration of Border Point Property

- DBSCAN assigns a border point to one cluster. The cluster to which it will be assigned depends on the order of processing points in dataset D:
- **Example.** Let minPts = 4. Point r belongs to $N_{Eps}(p)$ and $N_{Eps}(q)$, so r will be assigned either to the red cluster or to the blue one by DBSCAN.

33

Size of a DBSCAN Cluster

- It may happen that size of a cluster is less than minPts.
- **Example.** Let minPts = 4. Assume that point r, which belongs to $N_{Eps}(p)$ and $N_{Eps}(q)$, was assigned to the red cluster. Then the blue cluster will have only 3 points.



34

Major Challenges in DBSCAN

- Efficient calculation of Eps-neighborhood for each point.
- To this end, DBSCAN uses the R*-tree index.
- The use of such indices helps in the case of low dimensional data only.

35

TI-DBSCAN: DBSCAN with Efficient Calculation of Eps-Neighboorhoods

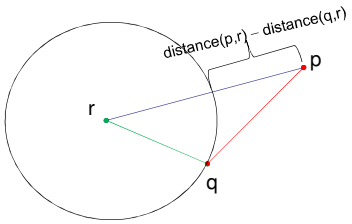
- Use the triangle inequality property (TI) to reduce the number of candidates for being a member of Eps-neighboorhood of a given point.

36

TI for pessimistic estimation of distance

For any three points p, q, r:

- distance(p,q) + distance(q,r) ≥ distance(p,r).
- **distance(p,q) ≥** distance(p,r) – distance(q,r).



37

TI & Eps-Neighborhood…

Lemma. Let D be a set of points. For any two points p, q in D and any point r:

distance(p,r) – distance(q,r) > Eps ⇒

distance(p,q) ≥ distance(p,r) – distance(q,r) > Eps ⇒

by TI

$q \notin N_{Eps}(p) \wedge p \notin N_{Eps}(q)$.

38

TI & Eps-Neighborhood…

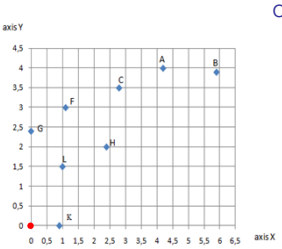
Theorem. Let:

- r be any point,
- D be a set of points ordered in a non-decreasing way wrt. their distances to r;
- p be any point in D;
- q be a point following point p in D such that distance(q,r) – distance(p,r) > Eps.

Then q and all points following q in D do not belong to $N_{Eps}(p)$.

39

Example: TI & Eps-Neighborhood…



Ordered set of points D; Eps = 0.2

q∈D	X	Y	distance(q,r)	
K	0,9	0,0	0,9	
L	1,0	1,5	1,8	
G	0,0	2,4	2,4	
H	2,4	2,0	3,1	
F	**1,1**	**3,0**	**3,2**	
C	2,8	3,5	4,5	∈ $N_{Eps}(F)$
A	4,2	4,0	5,8	
B	5,9	3,9	7,1	

40

TI & Eps-Neighborhood…

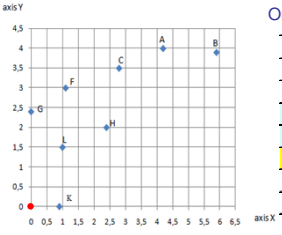
Theorem. Let:

- r be any point,
- D be a set of points ordered in a non-decreasing way wrt. their distances to r;
- p be any point in D;
- q be a point preceding point p in D such that distance(p,r) – distance(q,r) > Eps.

Then q and all points preceding q in D do not belong to $N_{Eps}(p)$.

41

Example: TI & Eps-Neighborhood…



Ordered set of points D; Eps = 0.2

q∈D	X	Y	distance(q,r)	
K	0,9	0,0	0,9	
L	1,0	1,5	1,8	
G	0,0	2,4	2,4	
H	2,4	2,0	3,1	
F	**1,1**	**3,0**	**3,2**	
C	2,8	3,5	4,5	
A	4,2	4,0	5,8	
B	5,9	3,9	7,1	

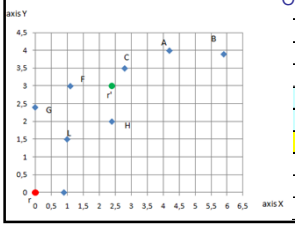
∈ $N_{Eps}(F)$

42

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Using Many Reference Points

Example. Let **r(0, 0)**, **r'(2.4, 3.0)**, Eps = 0.2. Then:
distance(F,r) – distance(H,r) = 3.2 – 3.1 = 0.1 ≤ Eps.
distance(F,r') – distance(H,r') = 1.3 – 1.0 = 0.3 > Eps.
Hence, H ∉ N_{Eps}(p).



Ordered set of points D; Eps = 0.2

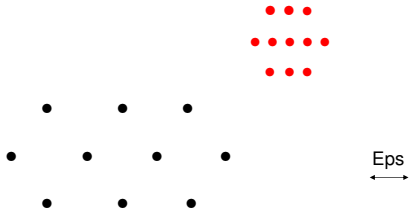
q∈D	X	Y	distance(q,r)	
K	0,9	0,0	0,9	
L	1,0	1,5	1,8	
G	0,0	2,4	2,4	
H	2,4	2,0	3,1	
F	**1,1**	**3,0**	**3,2**	∉ N_{Eps}(F)
C	2,8	3,5	4,5	
A	4,2	4,0	5,8	
B	5,9	3,9	7,1	

43

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Disadvantages of DBSCAN

- DBSCAN may not be able to discover clusters of different density.
- **Example.** Let minPts = 4. Only red cluster will be found.



44

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

NBC: Neighbourhood Clustering with Nearest Neighbours and Reversed Nearest Neighbours

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

NBC: Clustering Based on k⁺-neighbours and Reversed k⁺- neighbours

- *k⁺-neighbourhood of a point p* (k+NN(p)) is the set of all points in D that are distant from p by no more than any k-neighbour of p.
- A reversed *k⁺-neighbourhood of a point p* (Rk+NN(p)) is the set of all points in D having p as their k⁺-neighbour,

$$Rk^+NN(p) = \{q \in D | \ p \in k^+NN(q)\}.$$

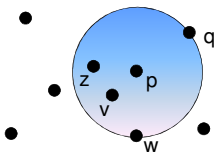
46

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: k⁺-neighbourhood and reversed k⁺-neighbourhood

Let k = 3. Then $|k^+\text{-NN}(p)| = 4$ and:

- Point q is a k⁺-neighbour of point p (i.e., q∈k⁺NN(p)).
- Point p is a reversed k⁺- neighbour of q (i.e., p∈Rk⁺NN(q)).



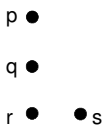
47

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: k⁺-neighbourhood and reversed k⁺- neighbourhood

Let k = 2. Then:

- k⁺-NN(p) = {q, r}; k⁺-NN(q) = {p, r}; k⁺-NN(r) = {q, s}; k⁺-NN(s) = {r, q}
- Rk⁺-NN(p) = {q}
- Rk⁺-NN(q) = {p,r,s}



48

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Clusters generated by NBC

- Density of a subspace is expressed by means of density factor NDF understood as the ratio of the cardinality of reversed k^+-neighbourhood to the cardinality of k^+-neighbourhood:

$$NDF(p) = \frac{|Rk^+\text{-}NN(p)|}{|k^+\text{-}NN(p)|}$$

- Point p plays a role of *a* core point if $NDF(p) \geq 1$.
- A core point is perceived as a seed that together with its k^+-neighbourhood represents a dense space, which can be regarded as a cluster or its part.

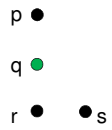
49

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: k^+-NN, Rk^+-NN and density factor NDF

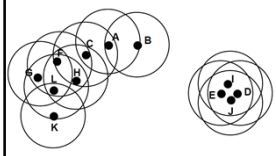
Let k = 2. Then:

- k^+-NN(p) = {q, r}; k^+-NN(q) = {p, r}; k^+-NN(r) = {q, s}; k^+-NN(s) = {r, q}
- Rk^+-NN(p) = {q}; NDF(p) = 1/2
- Rk^+-NN(q) = {p,r,s}; NDF(q) = 3/2



50

NBC – Calculation of k^+-NN and $|Rk^+$-NN| …

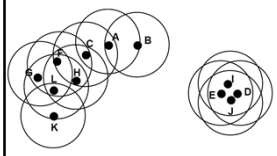


k = 3

q∈D	x	y	k^+-NN	$\|Rk^+$-NN$\|$	NDF	Cld
A	4.2	4.0	B, C, H	0		
B	5.9	3.9		~~0~~ 1		
C	2.8	3.5		~~0~~ 1		
D	12.0	1.3		0		
E	10.0	1.3		0		
F	1.1	3.0		0		
G	0.0	2.4		0		
H	2.4	2.0		~~0~~ 1		
I	11.5	1.8		0		
J	11.0	1.0		0		
K	0.9	0.0		0		
L	1.0	1.5		0		

51

NBC – Calculation of k^+-NN and $|Rk^+$-NN| …

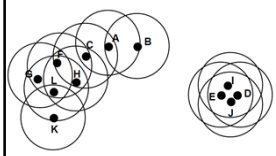


k = 3

q∈D	x	y	k^+-NN	$\|Rk^+$-NN$\|$	NDF	Cld
A	4.2	4.0	B, C, H	~~0~~ 1		
B	5.9	3.9	A, C, H	1		
C	2.8	3.5		~~1~~ 2		
D	12.0	1.3		0		
E	10.0	1.3		0		
F	1.1	3.0		0		
G	0.0	2.4		0		
H	2.4	2.0		~~1~~ 2		
I	11.5	1.8		0		
J	11.0	1.0		0		
K	0.9	0.0		0		
L	1.0	1.5		0		

52

NBC – Calculation of k^+-NN and $|Rk^+$-NN| …

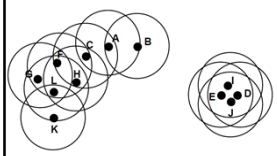


k = 3

q∈D	x	y	k^+-NN	$\|Rk^+$-NN$\|$	NDF	Cld
A	4.2	4.0	B, C, H	~~1~~ 2		
B	5.9	3.9	A, C, H	1		
C	2.8	3.5	A, F, H	2		
D	12.0	1.3		0		
E	10.0	1.3		0		
F	1.1	3.0		~~0~~ 1		
G	0.0	2.4		0		
H	2.4	2.0		~~2~~ 3		
I	11.5	1.8		0		
J	11.0	1.0		0		
K	0.9	0.0		0		
L	1.0	1.5		0		

53

NBC – Calculation of k^+-NN and $|Rk^+$-NN|

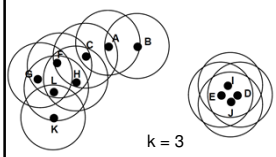


k = 3

q∈D	x	y	k^+-NN	$\|Rk^+$-NN$\|$	NDF	Cld
A	4.2	4.0	B, C, H	2		
B	5.9	3.9	A, C, H	1		
C	2.8	3.5	A, F, H	3		
D	12.0	1.3	E, I, J	3		
E	10.0	1.3	D, I, J	3		
F	1.1	3.0	G, H, L	~~3~~ 4		
G	0.0	2.4	F, H, L	~~2~~ 3		
H	2.4	2.0	C, F, L	~~6~~ 7		
I	11.5	1.8	D, E, J	3		
J	11.0	1.0	D, E, I	3		
K	0.9	0.0	G, H, L	~~0~~ 1		
L	1.0	1.5	F, G, H, K	4		

54

NBC – Calculation of NDF



k = 3

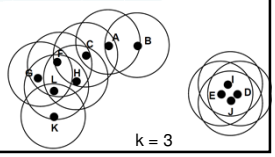
q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	
B	5.9	3.9	A, C, H	1	1/3	
C	2.8	3.5	A, F, H	3	3/3	
D	12.0	1.3	E, I, J	3	3/3	
E	10.0	1.3	D, I, J	3	3/3	
F	1.1	3.0	G, H, L	4	4/3	
G	0.0	2.4	F, H, L	3	3/3	
H	2.4	2.0	C, F, L	7	7/3	
I	11.5	1.8	D, E, J	3	3/3	
J	11.0	1.0	D, E, I	3	3/3	
K	0.9	0.0	G, H, L	1	1/3	
L	1.0	1.5	F, G, H, K	4	4/4	

55

NBC – Looking for first cluster…

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	N
B	5.9	3.9	A, C, H	1	1/3	
C	2.8	3.5	A, F, H	3	3/3	
D	12.0	1.3	E, I, J	3	3/3	
E	10.0	1.3	D, I, J	3	3/3	
F	1.1	3.0	G, H, L	4	4/3	
G	0.0	2.4	F, H, L	3	3/3	
H	2.4	2.0	C, F, L	7	7/3	
I	11.5	1.8	D, E, J	3	3/3	
J	11.0	1.0	D, E, I	3	3/3	
K	0.9	0.0	G, H, L	1	1/3	
L	1.0	1.5	F, G, H, K	4	4/4	

- Cld = 1.
- NDF(A) < 1, so A is non-core.

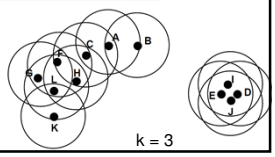


k = 3

NBC – Looking for first cluster…

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	N
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	
D	12.0	1.3	E, I, J	3	3/3	
E	10.0	1.3	D, I, J	3	3/3	
F	1.1	3.0	G, H, L	4	4/3	
G	0.0	2.4	F, H, L	3	3/3	
H	2.4	2.0	C, F, L	7	7/3	
I	11.5	1.8	D, E, J	3	3/3	
J	11.0	1.0	D, E, I	3	3/3	
K	0.9	0.0	G, H, L	1	1/3	
L	1.0	1.5	F, G, H, K	4	4/4	

- Cld = 1.
- NDF(B) < 1, so B is non-core.

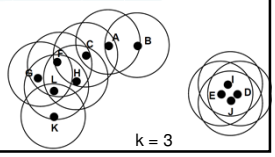


k = 3

NBC – Looking for first cluster

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	N 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	
E	10.0	1.3	D, I, J	3	3/3	
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	
J	11.0	1.0	D, E, I	3	3/3	
K	0.9	0.0	G, H, L	1	1/3	
L	1.0	1.5	F, G, H, K	4	4/4	

- Cld = 1.
- NDF(C) ≥ 1, so C is a core.
- Cluster 1 is initialized.
- seeds = {F, H}.

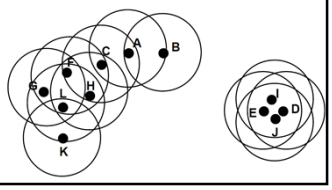


k = 3

NBC – Extension of a Cluster

- Dealing with a point q from an analysed k⁺-NN:

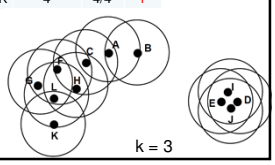
if q.ClusterId = Unclassified **then** // q is not classified
  assign q *to currently created cluster*
  add q *to seeds*
else if q.ClusterId = N // q is not a core
  assign q *to currently created cluster*
else
  do nothing



NBC – Extending first cluster…

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	N 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	
E	10.0	1.3	D, I, J	3	3/3	
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	
J	11.0	1.0	D, E, I	3	3/3	
K	0.9	0.0	G, H, L	1	1/3	
L	1.0	1.5	F, G, H, K	4	4/4	1

- Cld = 1.
- seeds = {F, H}.
- NDF(F) ≥ 1, so F is a core.
- Updated seeds = {H, G, L}.

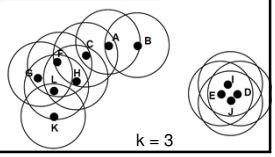


k = 3

NBC – Extending first cluster…

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	~~N~~ 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	
E	10.0	1.3	D, I, J	3	3/3	
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	
J	11.0	1.0	D, E, I	3	3/3	
K	0.9	0.0	G, H, L	1	1/3	
L	1.0	1.5	F, G, H, K	4	4/4	1

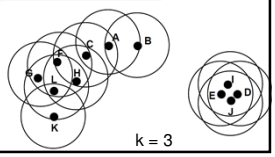
- Cld = 1.
- seeds = {H, G, L}
- NDF(H) ≥ 1, so H is a core.
- Updated seeds = {G, L}.



NBC – Extending first cluster…

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	~~N~~ 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	
E	10.0	1.3	D, I, J	3	3/3	
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	
J	11.0	1.0	D, E, I	3	3/3	
K	0.9	0.0	G, H, L	1	1/3	
L	1.0	1.5	F, G, H, K	4	4/4	1

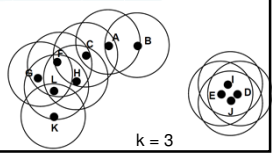
- Cld = 1.
- seeds = {G, L}
- NDF(G) ≥ 1, so G is a core.
- Updated seeds = {L}.



NBC – Extending first cluster…

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	~~N~~ 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	
E	10.0	1.3	D, I, J	3	3/3	
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	
J	11.0	1.0	D, E, I	3	3/3	
K	0.9	0.0	G, H, L	1	1/3	1
L	1.0	1.5	F, G, H, K	4	4/4	1

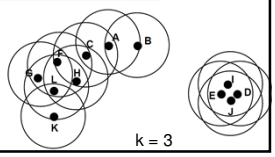
- Cld = 1.
- seeds = {L}
- NDF(L) ≥ 1, so L is a core.
- Updated seeds = {K}.



NBC – Extending first cluster

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	~~N~~ 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	
E	10.0	1.3	D, I, J	3	3/3	
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	
J	11.0	1.0	D, E, I	3	3/3	
K	0.9	0.0	G, H, L	1	1/3	1
L	1.0	1.5	F, G, H, K	4	4/4	1

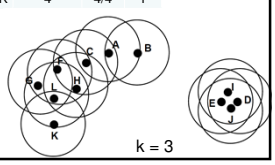
- Cld = 1.
- seeds = {K}
- NDF(K) < 1, so K is non-core.
- Updated seeds = {}.
- Cluster 1 was created.



NBC – Looking for second cluster

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	~~N~~ 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	2
E	10.0	1.3	D, I, J	3	3/3	2
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	2
J	11.0	1.0	D, E, I	3	3/3	2
K	0.9	0.0	G, H, L	1	1/3	1
L	1.0	1.5	F, G, H, K	4	4/4	1

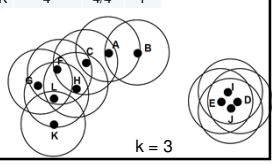
- Cld = 2.
- NDF(D) ≥ 1, so D is a core.
- Cluster 2 is initialized.
- seeds = {E, I, J}.



NBC – Extending second cluster…

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	~~N~~ 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	2
E	10.0	1.3	D, I, J	3	3/3	2
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	2
J	11.0	1.0	D, E, I	3	3/3	2
K	0.9	0.0	G, H, L	1	1/3	1
L	1.0	1.5	F, G, H, K	4	4/4	1

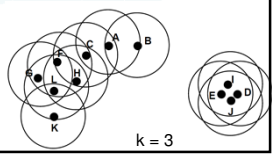
- Cld = 2.
- seeds = {E, I, J}.
- NDF(E) ≥ 1, so E is a core.
- Updated seeds = {I, J}.



NBC – Extending second cluster…

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	~~N~~ 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	2
E	10.0	1.3	D, I, J	3	3/3	2
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	2
J	11.0	1.0	D, E, I	3	3/3	2
K	0.9	0.0	G, H, L	1	1/3	1
L	1.0	1.5	F, G, H, K	4	4/4	1

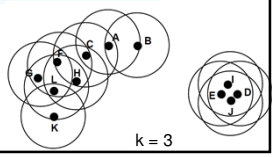
- Cld = 2.
- seeds = {I, J}.
- NDF(I) ≥ 1, so I is a core.
- Updated seeds = {J}.



NBC – Extending second cluster

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	~~N~~ 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	2
E	10.0	1.3	D, I, J	3	3/3	2
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	2
J	11.0	1.0	D, E, I	3	3/3	2
K	0.9	0.0	G, H, L	1	1/3	1
L	1.0	1.5	F, G, H, K	4	4/4	1

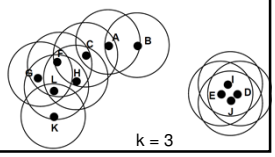
- Cld = 2.
- seeds = {J}.
- NDF(J) ≥ 1, so J is a core.
- Updated seeds = {}.
- Cluster 2 was created.



NBC – Result of Clustering

q∈D	x	y	k⁺-NN	|Rk⁺-NN|	NDF	Cld
A	4.2	4.0	B, C, H	2	2/3	~~N~~ 1
B	5.9	3.9	A, C, H	1	1/3	N
C	2.8	3.5	A, F, H	3	3/3	1
D	12.0	1.3	E, I, J	3	3/3	2
E	10.0	1.3	D, I, J	3	3/3	2
F	1.1	3.0	G, H, L	4	4/3	1
G	0.0	2.4	F, H, L	3	3/3	1
H	2.4	2.0	C, F, L	7	7/3	1
I	11.5	1.8	D, E, J	3	3/3	2
J	11.0	1.0	D, E, I	3	3/3	2
K	0.9	0.0	G, H, L	1	1/3	1
L	1.0	1.5	F, G, H, K	4	4/4	1

- All points in dataset D were processed, so the clustering is done.
- The obtained clustering result: 2 clusters and 1 noise point (here: point B).
- Cluster 1 has 2 border points (A, K).
- Cluster 2 does not have border points.



WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

TI-NBC: NBC Clustering with Efficient Calculation of k+-neighboorhood

By applying:

- multiple estimation of a decreasing radius within which k+-neighbourhood is guaranteed to be found
- triangle inequality property (TI)

in order to reduce the number of distance calculations.

70

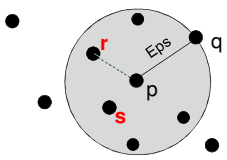
Estimation of the radius of the smallest $N_{Eps}(p)$ that contains k-NN(p) (k⁺-NN(p))

- **Property:** Let Eps denote the greatest distance from p to some other k points. Then:

$$k\text{-NN}(p) \subseteq k^{+}\text{–NN}(p) \subseteq N_{Eps}(p).$$

- **Example.** Let k = 2 and the distances from point p to the (arbitrarily chosen) points r and q have already been calculated. Let Eps equal the greater of these distances. Then:

$$k\text{-NN}(p) = k^{+}\text{–NN}(p) = \{r, s\} \subseteq N_{Eps}(p)$$

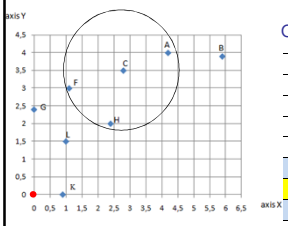


WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

TI in Searching k⁺-NN…

Example. Let **r(0, 0)**, k = 2 and first k candidates for k⁺-NN of point C are those with the least pessimistic estimation of its distances to C, that is, *Candidates* = {A, F}.

- Then, Eps = max(distance(C,A), distance(C,F)) = max{1.49, 1.77} = 1.77.



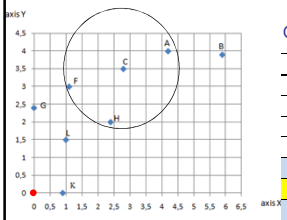
Ordered set of points D

q	X	Y	distance(q,r)	distance(C,q)
K	0,9	0,0	0,9	
L	1,0	1,5	1,8	
G	0,0	2,4	2,4	
H	2,4	2,0	3,1	
F	1,1	3,0	3,2	1.77
C	2,8	3,5	4,5	
A	4,2	4,0	5,8	1.49
B	5,9	3,9	7,1	

TI in Searching k⁺-NN…

Example. **r(0, 0)**, k = 2. *Candidates* = {A, F}.

- distance(B,r) - distance(C,r) = 2.6 > Eps = 1.77.
- So, distance(B,C) > Eps = 1.77 and B ∉ k⁺-NN(C).



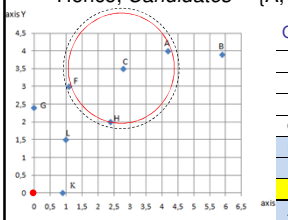
Ordered set of points D; Eps = 1.77

q	X	Y	distance(q,r)	distance(C,q)
K	0,9	0,0	0,9	
L	1,0	1,5	1,8	
G	0,0	2,4	2,4	
H	2,4	2,0	3,1	
F	1,1	3,0	3,2	1.77
C	2,8	3,5	4,5	
A	4,2	4,0	5,8	1.49
B	5,9	3,9	7,1	

TI in Searching k⁺-NN…

Example. Let **r(0, 0)**, k = 2. *Candidates* = {A, F}.

- distance(C,r) - distance(H,r) = 1.4 < Eps = 1.77.
- distance(C,H) = 1.55 < Eps = 1.77.
- Hence, *Candidates'* = {A, H} and Eps' = 1.55.



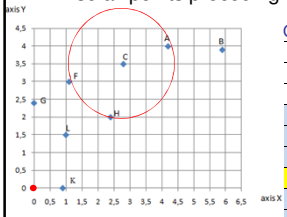
Ordered set of points D; Eps = 1.77

q	X	Y	distance(q,r)	distance(C,q)
K	0,9	0,0	0,9	
L	1,0	1,5	1,8	
G	0,0	2,4	2,4	
H	2,4	2,0	3,1	1.55
F	1,1	3,0	3,2	1.77
C	2,8	3,5	4,5	
A	4,2	4,0	5,8	1.49
B	5,9	3,9	7,1	

TI in Searching k⁺-NN

Example. Let **r(0, 0)**, k = 2. Candidates' = {A, H}.

- distance(C,r) - distance(G,r) = 2.1 > Eps' = 1.55.
- So, distance(C,G) > Eps' = 1.55 and G ∉ k⁺-NN(C).
- Also all points preceding G do not belong to k⁺-NN(C).



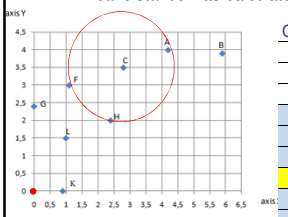
Ordered set of points D; Eps' = 1.55

q	X	Y	distance(q,r)	distance(C,q)
K	0,9	0,0	0,9	
L	1,0	1,5	1,8	
G	0,0	2,4	2,4	
H	2,4	2,0	3,1	1.55
F	1,1	3,0	3,2	1.77
C	2,8	3,5	4,5	
A	4,2	4,0	5,8	1.49
B	5,9	3,9	7,1	

TI in Searching k⁺-NN

Example. Let **r(0, 0)**, k = 2. Candidates' = {A, H}.

- Summary of looking for k⁺-NN(C):
  - k⁺-NN = {A, H}.
  - Pessimistic estimation of distance was calculated for 5 points.
  - Real distance was calculated for 3 points.



Ordered set of points D; Eps' = 1.55

q	X	Y	distance(q,r)	distance(C,q)
K	0,9	0,0	0,9	
L	1,0	1,5	1,8	
G	0,0	2,4	2,4	
H	2,4	2,0	3,1	1.55
F	1,1	3,0	3,2	1.77
C	2,8	3,5	4,5	
A	4,2	4,0	5,8	1.49
B	5,9	3,9	7,1	

Scaling of Attribute Values

Scaling of Continuous Attributes: range

Let:

- v be a value of a continuous attribute A,
- v_{min} be the least value of A,
- v_{max} be the greatest value of A.

Then:

$$range(v) = \frac{v - v_{min}}{v_{max} - v_{min}}.$$

78

Scaling of Continuous Attributes: Z-score

Let D consist of *n* data points that have values $v_1, \ldots, v_n$ for continuous attribute A. Then:

$$Z - score(v) = \frac{v-\mu}{s}, \text{ where}$$

- the mean for A:

$$\mu = \frac{1}{n}(v_1 + \cdots + v_n),$$

- the mean absolute deviation for A:

$$S = \frac{1}{n}(|v_1 - \mu| + \cdots + |v_n - \mu|).$$

79

Quality of Clustering

Evaluation of Clustering

- Evaluation based on external information: calculated clusters can be compared with real clusters (e.g. determined by a knowledgeable user).
- Evaluation based on internal information.

81

External Evaluation of Clustering with Purity

$$Purity = \frac{1}{n}\sum_{g \in G} max_{c \in C} |g \cap c|, \text{ where}$$

C – real clusters,

G – discovered clusters,

n – the number of points.

82

Example: External Evaluation of Clustering with Purity

Real clusters	Discovered clusters	Correct assignment of points to clusters
L	1	
L	3	
L	2	Yes (L)
L	2	Yes (L)
L	2	Yes (L)
H	2	
H	1	Yes (H)
H	1	Yes (H)
H	3	Yes (H)
H	3	Yes (H)

Purity = 7/10

83

External Evaluation of Clustering with Rand

$$Rand = \frac{|TP| + |TN|}{\binom{n}{2}}, \text{ where}$$

- TP – the set of pairs of objects that are in the same real cluster and in the same discovered cluster,
- TN – the set of pairs of objects that are in different real clusters and in different discovered clusters,
- *n* – the number of objects.

84

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: External Evaluation of Clustering with Rand

Id	Real clusters	Discovered clusters
1	L	1
2	L	3
3	L	2
4	L	2
5	L	2
6	H	2
7	H	1
8	H	1
9	H	3
10	H	3

- Pairs of objects the set of pairs of objects that are in the same real cluster and in the same discovered cluster:

  TP = {(3,4), (3,5), (4,5), (7,8), (9,10).}

- Pairs of objects the set of pairs of objects that are in different real clusters and in different discovered clusters:

  TN = {(1,6), (1,9), (1,10), (2,6), (2,7), (2,8), (3,7), (3,8), (3,9), (3,10), (4,7), (4,8), (4,9), (4,10), (5,7), (5,8), (5,9), (5,10)}.

$Rand = \frac{|TP| + |TN|}{\binom{10}{2}} = \frac{5+18}{45} \approx 0.51$

85

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Internal Evaluation of Clustering with Davies-Bouldin

$Davies\text{–}Bouldin = \frac{1}{n}\sum_{i=1}^{n} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$, where

- n – the number of discovered clusters,
- c_k – the centroid of cluster k,
- σ_k – the average distance of points in cluster k *to its* centroid c_k,
- $d(c_i, c_j)$ – the distance between centroids c_i, c_j.

86

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Internal Evaluation of Clustering with Silhouette Coefficient

Quality of assigning a point i to its cluster:

$$s(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}}, \text{ where}$$

- $a(i)$ – the average distance of point i to all other points in its cluster,
- $b(i)$ – the least average distance of point i to all points of a cluster that does not contain point i.

Quality of a cluster C – the average $s(i)$ over all points i in C.

Quality of clustering – the average $s(i)$ over all points i in the whole data set.

87

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

References…

- Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD 1996: 226-231
- Jiawei Han, Micheline Kamber, Jian Pei: Data Mining: Concept and Techniques, The Morgan Kaufmann Series in Data Management Systems, 2011

88

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

References…

- Marzena Kryszkiewicz, Bartlomiej Janczak: Basic Triangle Inequality Approach Versus Metric VP-Tree and Projection in Determining Euclidean and Cosine Neighbors. Intelligent Tools for Building a Scientific Information Platform 2014: 27-49
- Marzena Kryszkiewicz, Piotr Lasek: TI-DBSCAN: Clustering with DBSCAN by Means of the Triangle Inequality. RSCTC 2010: 60-69

89

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

References…

- Marzena Kryszkiewicz, Piotr Lasek: A Neighborhood-Based Clustering by Means of the Triangle Inequality. IDEAL 2010: 284-291
- Shuigeng Zhou, Yue Zhao, Jihong Guan, Joshua Zhexue Huang: A Neighborhood-Based Clustering Algorithm. PAKDD 2005: 361-371

90

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

References

- https://en.wikipedia.org/wiki/Cluster_analysis
- https://en.wikipedia.org/wiki/Silhouette_(clustering)

91