Part 2 - 03: Urban Data Mining (Spatial Data)

Long Cheng

Assistant Professor

c.long@ntu.edu.sg

Spatial Data Mining

Spatial Data Querying

- Point query
- Range query
- Nearest neighbor
- Spatial join
- Spatial matching

Can not answer complex questions

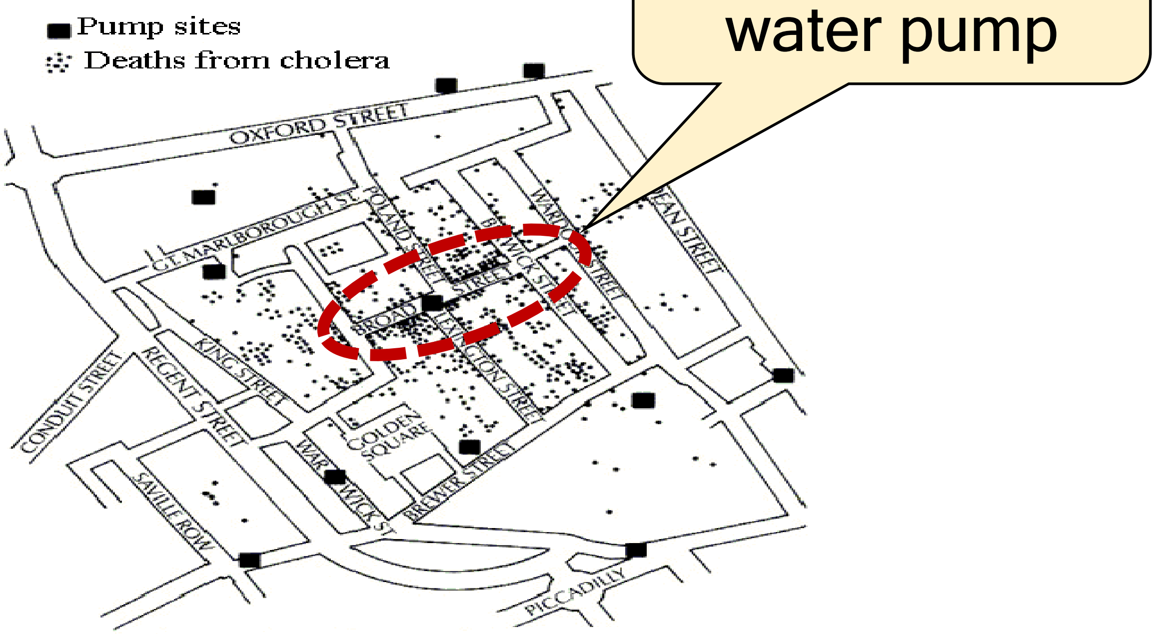
Spatial Data Mining

- Spatial clustering (Hotspot)
- Spatial outlier detection
- Co-location mining

Spatial Data Mining

Spatial clustering (Hotspot)	Spatial outlier detection	Co-location mining

Spatial Clustering: Example - Hotspots



The 1854 Asiatic Cholera in London

Spatial Clustering: Methods

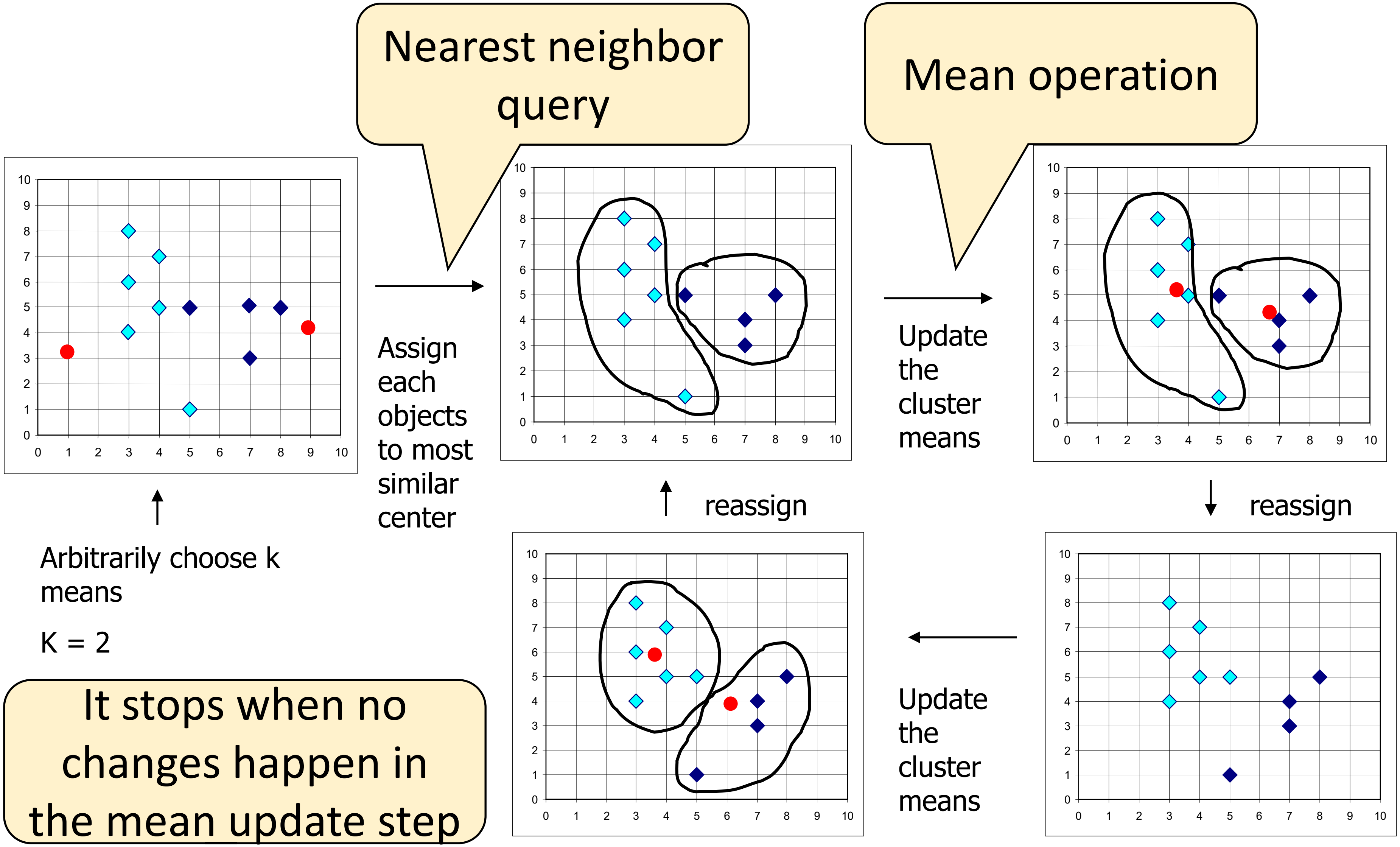
K-means	Hierarchical Clustering	EM Clustering	DBSCAN

K-Means Clustering

K-Means Clustering:

1. Make initial guesses for the means m_1, m_2, .., m_k
2. Until there is no change in any mean
   - **(Re)assign** each data point to the cluster whose mean is the nearest
   - **Update** m_i with the mean of all examples for each cluster i (i = 1, 2, …, k)

K-Means Clustering



K-Means Clustering

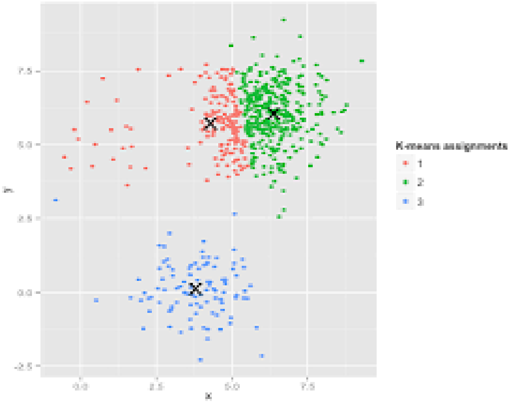
Notes

- The way to initialize the means was not specified. One popular way to start is to randomly choose k of the examples
- The results produced depend on the initial values of the means, and it frequently happens that suboptimal partitions are found. The standard solution is to try a number of different starting points

K-Means Clustering

Disadvantages

- In a “bad” initial guess, there are no points assigned to the cluster with the initial mean m_i.
- The value of k is not user-friendly. This is because we do not know the number of clusters before we want to find clusters.



Spatial Clustering: Methods

K-means	Hierarchical Clustering	EM Clustering	DBSCAN

Hierarchical Clustering

Hierarchical Clustering:

- The clusters are computed recursively via multiple steps.
- There are two varieties of hierarchical clustering algorithms
  - Agglomerative – successively fusions of the data into groups
  - Divisive – separate the data successively into finer groups

Hierarchical Clustering

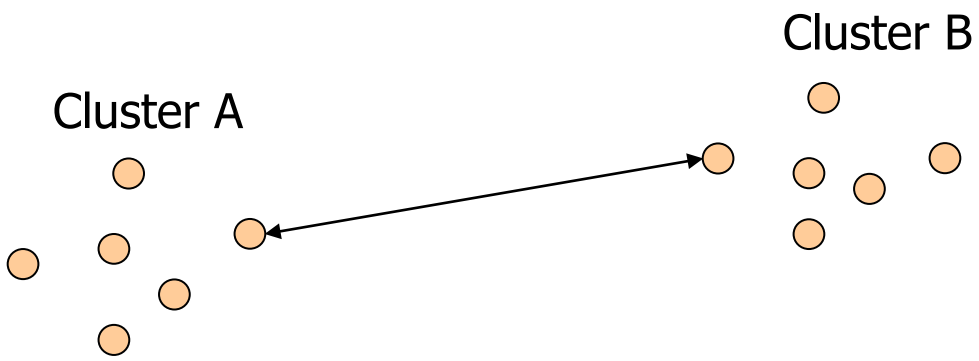
Distance (between two clusters)

1. Single Linkage
2. Complete Linkage
3. Group Average Linkage
4. Centroid Linkage
5. Median Linkage

Single Linkage

Single Linkage

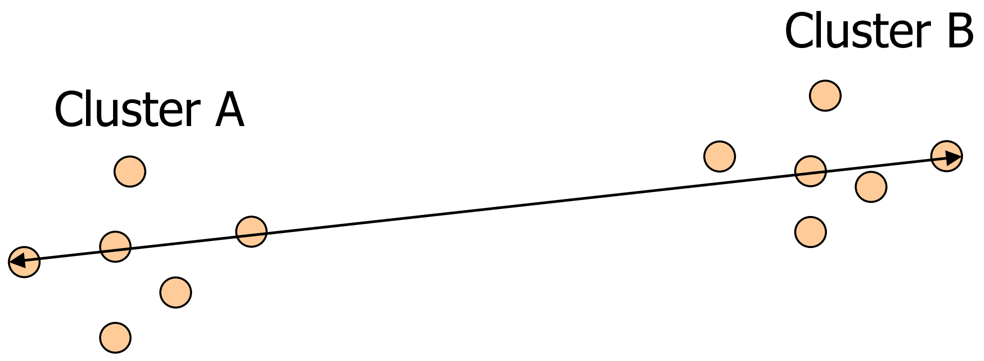
- Also, known as the **nearest neighbor** technique
- Distance between groups is defined as that of the closest pair of data, where only pairs consisting of one record from each group are considered



Complete Linkage

Complete Linkage

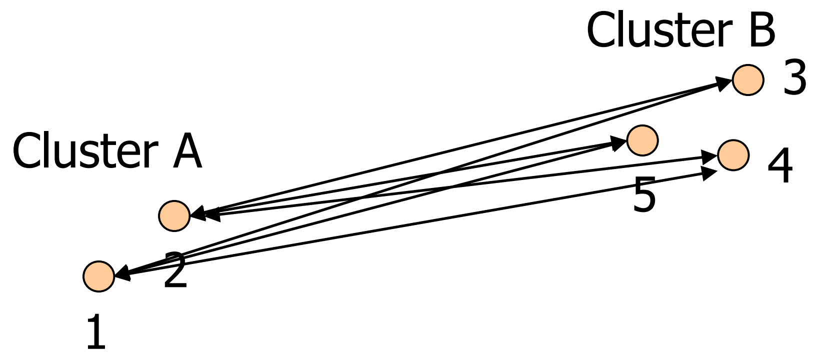
- The distance between two clusters is given by the distance between their most distant members



Group Average Linkage

Group Average Linkage

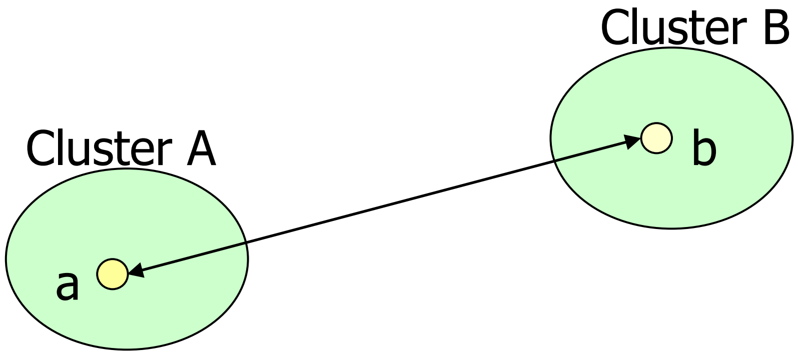
- The distance between two clusters is defined as the average of the distances between all pairs of records (one from each cluster).
- $d_{AB} = 1/6\ (d_{13} + d_{14} + d_{15} + d_{23} + d_{24} + d_{25})$



Centroid Linkage

Centroid Linkage

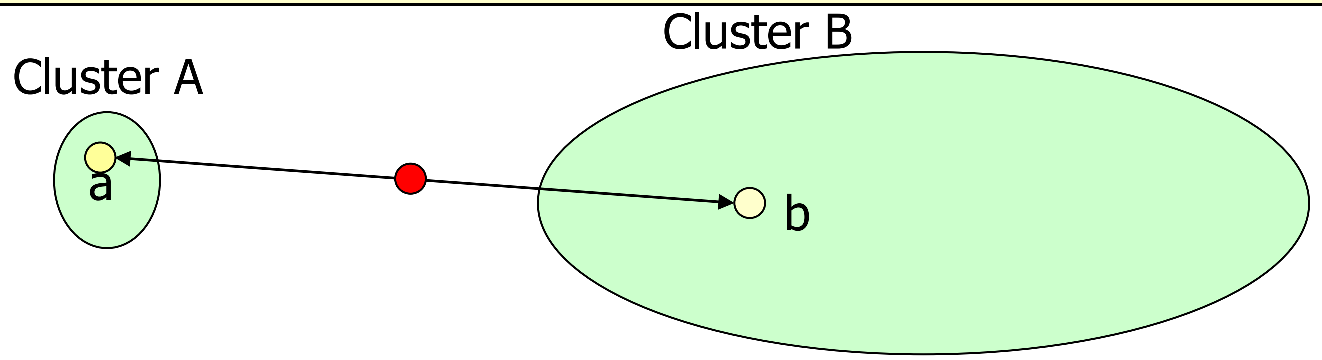
- The distance between two clusters is defined as the distance between the mean vectors of the two clusters.
- $d_{AB} = d_{ab}$
- where a is the mean vector of the cluster A and b is the mean vector of the cluster B.



Median Linkage

Median Linkage

- Disadvantage of the Centroid Clustering: When a large cluster is merged with a small one, the centroid of the combined cluster would be closed to the large one, ie. The characteristic properties of the small one are lost
- After we have combined two groups, the mid-point of the original two cluster centres is used as the centre of the newly combined group



Hierarchical Clustering

Hierarchical Clustering:

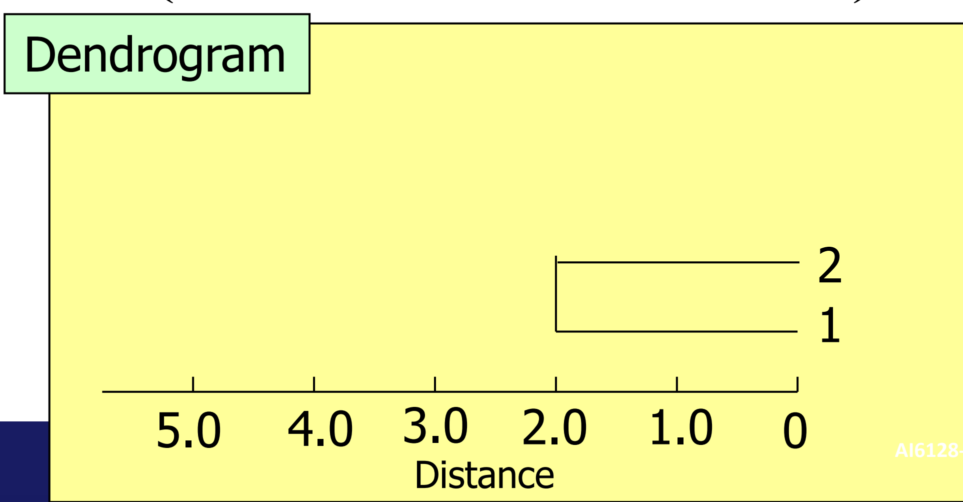
- The clusters are computed recursively via multiple steps.
- There are two varieties of hierarchical clustering algorithms
  - Agglomerative – successively fusions of the data into groups
  - Divisive – separate the data successively into finer gro

Assuming **single linkage**

Agglomerative Clustering

$$
\begin{array}{c}
 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{pmatrix}
0.0 & & & & \\
2.0 & 0.0 & & & \\
6.0 & 5.0 & 0.0 & & \\
10.0 & 9.0 & 4.0 & 0.0 & \\
9.0 & 8.0 & 5.0 & 3.0 & 0.0
\end{pmatrix}
\end{array}
$$

$$
\begin{array}{c|cccc}
 & (12) & 3 & 4 & 5 \\
\hline
(12) & 0.0 & & & \\
3 & 5.0 & 0.0 & & \\
4 & 9.0 & 4.0 & 0.0 & \\
5 & 8.0 & 5.0 & 3.0 & 0.0
\end{array}
$$

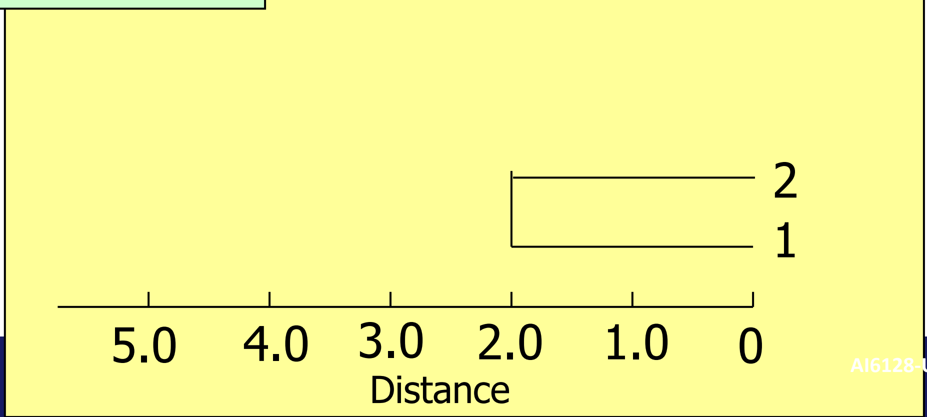


Agglomerative Clustering

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{pmatrix}
0.0 & & & & \\
2.0 & 0.0 & & & \\
6.0 & 5.0 & 0.0 & & \\
10.0 & 9.0 & 4.0 & 0.0 & \\
9.0 & 8.0 & 5.0 & 3.0 & 0.0
\end{pmatrix}
\end{array}
$$

$$
\begin{array}{c|c}
 & \begin{array}{cccc} (12) & 3 & 4 & 5 \end{array} \\
\begin{array}{c} (12) \\ 3 \\ 4 \\ 5 \end{array} &
\begin{pmatrix}
0.0 & & & \\
5.0 & 0.0 & & \\
9.0 & 4.0 & 0.0 & \\
8.0 & 5.0 & 3.0 & 0.0
\end{pmatrix}
\end{array}
$$

Dendrogram

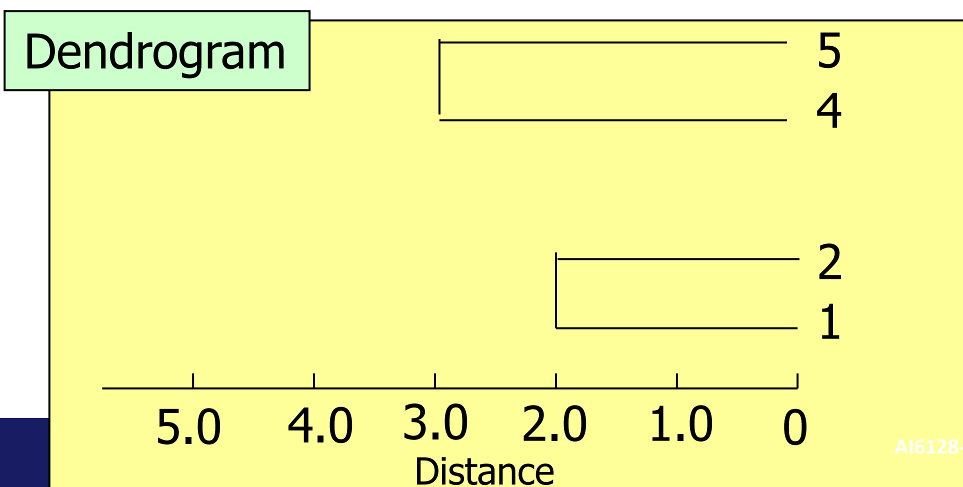


Agglomerative Clustering

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0.0 & & & & \\ 2 & 2.0 & 0.0 & & & \\ 3 & 6.0 & 5.0 & 0.0 & & \\ 4 & 10.0 & 9.0 & 4.0 & 0.0 & \\ 5 & 9.0 & 8.0 & 5.0 & 3.0 & 0.0 \end{array}$$

$$\begin{array}{c|cccc} & (12) & 3 & 4 & 5 \\ \hline (12) & 0.0 & & & \\ 3 & 5.0 & 0.0 & & \\ 4 & 9.0 & 4.0 & 0.0 & \\ 5 & 8.0 & 5.0 & 3.0 & 0.0 \end{array}$$

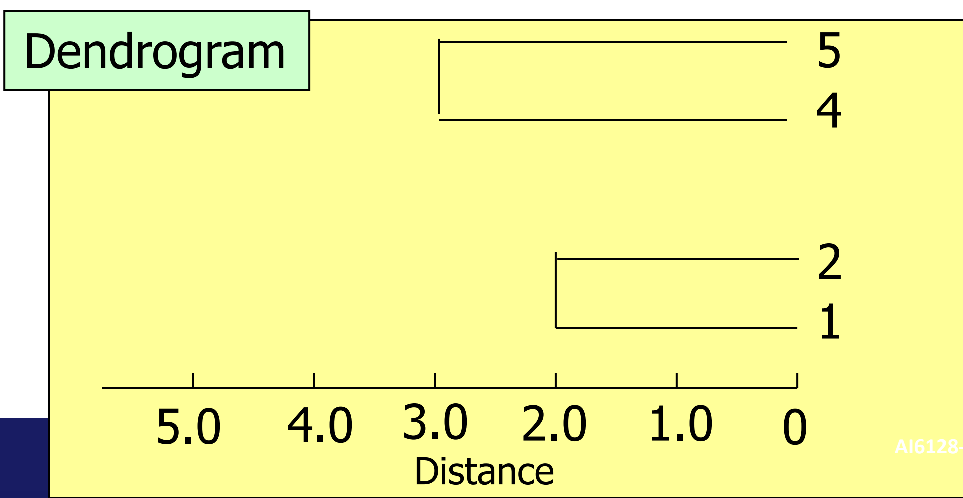
$$\begin{array}{c|ccc} & (12) & 3 & (4\ 5) \\ \hline (12) & 0.0 & & \\ 3 & 5.0 & 0.0 & \\ (4\ 5) & 8.0 & 4.0 & 0.0 \end{array}$$



Agglomerative Clustering

$$
\begin{array}{c c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 0.0 & & & & \\ 2.0 & 0.0 & & & \\ 6.0 & 5.0 & 0.0 & & \\ 10.0 & 9.0 & 4.0 & 0.0 & \\ 9.0 & 8.0 & 5.0 & 3.0 & 0.0 \end{pmatrix}
\end{array}
$$

$$
\begin{array}{c c}
 & \begin{array}{ccc} (12) & 3 & (4\ 5) \end{array} \\
\begin{array}{c} (12) \\ 3 \\ (4\ 5) \end{array} & \begin{pmatrix} 0.0 & & \\ 5.0 & 0.0 & \\ 8.0 & 4.0 & 0.0 \end{pmatrix}
\end{array}
$$

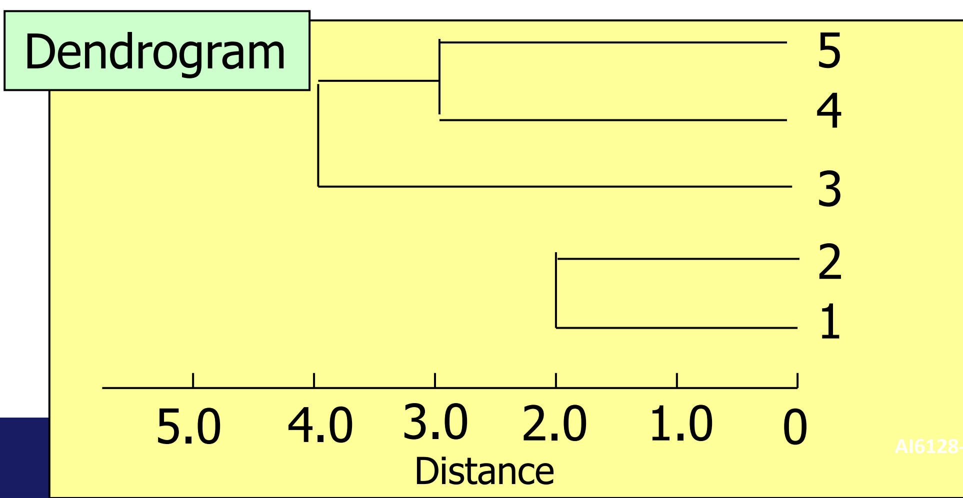


Agglomerative Clustering

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 0.0 & & & & \\ 2.0 & 0.0 & & & \\ 6.0 & 5.0 & 0.0 & & \\ 10.0 & 9.0 & 4.0 & 0.0 & \\ 9.0 & 8.0 & 5.0 & 3.0 & 0.0 \end{pmatrix} \end{array}$$

$$\begin{array}{cc} & \begin{array}{ccc} (12) & 3 & (4\ 5) \end{array} \\ \begin{array}{c} (12) \\ 3 \\ (4\ 5) \end{array} & \begin{pmatrix} 0.0 & & \\ 5.0 & 0.0 & \\ 8.0 & 4.0 & 0.0 \end{pmatrix} \end{array}$$

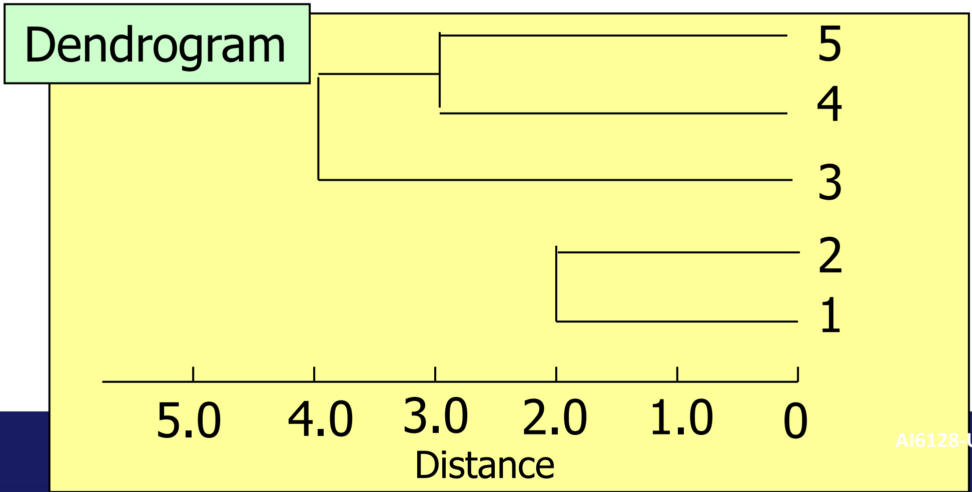
$$\begin{array}{cc} & \begin{array}{cc} (12) & (3\ 4\ 5) \end{array} \\ \begin{array}{c} (12) \\ (3\ 4\ 5) \end{array} & \begin{pmatrix} 0.0 & \\ 5.0 & 0.0 \end{pmatrix} \end{array}$$



Agglomerative Clustering

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0.0 & & & & \\ 2 & 2.0 & 0.0 & & & \\ 3 & 6.0 & 5.0 & 0.0 & & \\ 4 & 10.0 & 9.0 & 4.0 & 0.0 & \\ 5 & 9.0 & 8.0 & 5.0 & 3.0 & 0.0 \end{array}$$

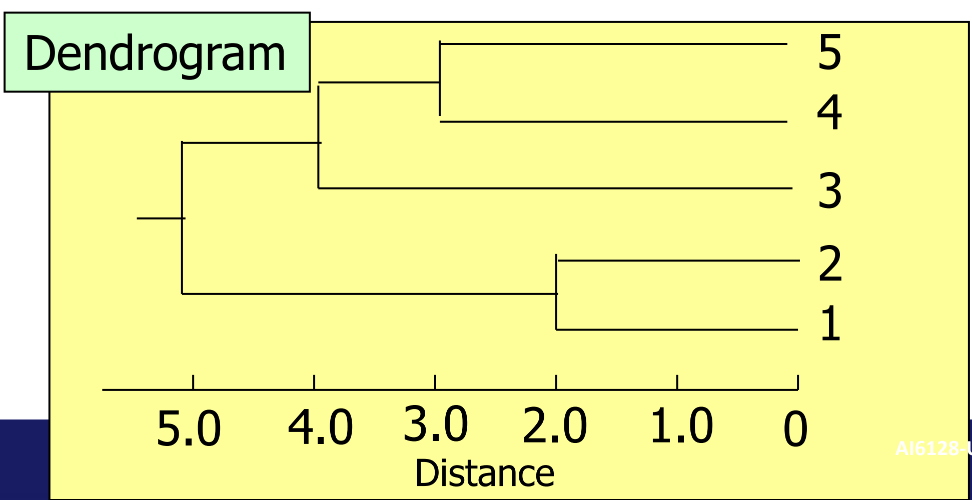
$$\begin{array}{c|cc} & (12) & (3\ 4\ 5) \\ \hline (12) & 0.0 & \\ (3\ 4\ 5) & 5.0 & 0.0 \end{array}$$



Agglomerative Clustering

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0.0 & & & & \\ 2 & 2.0 & 0.0 & & & \\ 3 & 6.0 & 5.0 & 0.0 & & \\ 4 & 10.0 & 9.0 & 4.0 & 0.0 & \\ 5 & 9.0 & 8.0 & 5.0 & 3.0 & 0.0 \end{array}$$

$$\begin{array}{c|cc} & (12) & (3\ 4\ 5) \\ \hline (12) & 0.0 & \\ (3\ 4\ 5) & 5.0 & 0.0 \end{array}$$



Hierarchical Clustering

Hierarchical Clustering:

- The clusters are computed recursively via multiple steps.
- There are two varieties of hierarchical clustering algorithms
  - Agglomerative – successively fusions of the data into groups
  - Divisive – separate the data successively into finer gro

Group Average Linkage

Divisive Clustering

Divisive Clustering

- In a divisive algorithm, we start with the assumption that all the data is part of one cluster.
- We then use a distance criterion to divide the cluster in two, and then subdivide the clusters until a stopping criterion is achieved.

Divisive Clustering

$$
\begin{array}{c c}
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\begin{pmatrix}
0 & & & & & & \\
10 & 0 & & & & & \\
7 & 7 & 0 & & & & \\
30 & 23 & 21 & 0 & & & \\
29 & 25 & 22 & 7 & 0 & & \\
38 & 34 & 31 & 10 & 11 & 0 & \\
42 & 36 & 36 & 13 & 17 & 9 & 0
\end{pmatrix}
\end{array}
$$

D(1, *) = 26.0

D(2, *) = 22.5

D(3, *) = 20.7

D(4, *) = 17.3

D(5, *) = 18.5

D(6, *) = 22.2

D(7, *) = 25.5

A = {1 }

B = {2, 3, 4, 5, 6, 7}

Divisive Clustering

$$
\begin{array}{c|c}
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\begin{pmatrix}
0 & & & & & & \\
10 & 0 & & & & & \\
7 & 7 & 0 & & & & \\
30 & 23 & 21 & 0 & & & \\
29 & 25 & 22 & 7 & 0 & & \\
38 & 34 & 31 & 10 & 11 & 0 & \\
42 & 36 & 36 & 13 & 17 & 9 & 0
\end{pmatrix}
\end{array}
$$

D(2, A) = 10

D(3, A) = 7

D(4, A) = 30

D(5, A) = 29

D(6, A) = 38

D(7, A) = 42

A = {1 }

B = {2, 3, 4, 5, 6, 7}

Divisive Clustering

$$
\begin{array}{c|c}
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\begin{pmatrix}
0 & & & & & & \\
10 & 0 & & & & & \\
7 & 7 & 0 & & & & \\
30 & 23 & 21 & 0 & & & \\
29 & 25 & 22 & 7 & 0 & & \\
38 & 34 & 31 & 10 & 11 & 0 & \\
42 & 36 & 36 & 13 & 17 & 9 & 0
\end{pmatrix}
\end{array}
$$

D(2, A) = 10	D(2, B) = 25.0
D(3, A) = 7	D(3, B) = 23.4
D(4, A) = 30	D(4, B) = 14.8
D(5, A) = 29	D(5, B) = 16.4
D(6, A) = 38	D(6, B) = 19.0
D(7, A) = 42	D(7, B) = 22.2

A = {1 }

B = {2, 3, 4, 5, 6, 7}

Divisive Clustering

$$
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & & & & & & \\
2 & 10 & 0 & & & & & \\
3 & 7 & 7 & 0 & & & & \\
4 & 30 & 23 & 21 & 0 & & & \\
5 & 29 & 25 & 22 & 7 & 0 & & \\
6 & 38 & 34 & 31 & 10 & 11 & 0 & \\
7 & 42 & 36 & 36 & 13 & 17 & 9 & 0
\end{array}
$$

D(2, A) = 10	D(2, B) = 25.0	Δ_2 = 15.0
D(3, A) = 7	D(3, B) = 23.4	Δ_3 = 16.4
D(4, A) = 30	D(4, B) = 14.8	Δ_4 = -15.2
D(5, A) = 29	D(5, B) = 16.4	Δ_5 = -12.6
D(6, A) = 38	D(6, B) = 19.0	Δ_6 = -19.0
D(7, A) = 42	D(7, B) = 22.2	Δ_7 = -19.8

$A = \{1, 3 \}$

$B = \{2, \cancel{3}, 4, 5, 6, 7\}$

Divisive Clustering

$$
\begin{array}{c|c}
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\begin{pmatrix}
0 & & & & & & \\
10 & 0 & & & & & \\
7 & 7 & 0 & & & & \\
30 & 23 & 21 & 0 & & & \\
29 & 25 & 22 & 7 & 0 & & \\
38 & 34 & 31 & 10 & 11 & 0 & \\
42 & 36 & 36 & 13 & 17 & 9 & 0
\end{pmatrix}
\end{array}
$$

$A = \{1, 3 \quad \}$

$B = \{2, \quad 4, 5, 6, 7\}$

$D(2, A) = 10$	$D(2, B) = 25.0$	$\Delta_2 = 15.0$
$D(3, A) = 7$	$D(3, B) = 23.4$	$\Delta_3 = 16.4$
$D(4, A) = 30$	$D(4, B) = 14.8$	$\Delta_4 = -15.2$
$D(5, A) = 29$	$D(5, B) = 16.4$	$\Delta_5 = -12.6$
$D(6, A) = 38$	$D(6, B) = 19.0$	$\Delta_6 = -19.0$
$D(7, A) = 42$	$D(7, B) = 22.2$	$\Delta_7 = -19.8$

Divisive Clustering

$$
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & & & & & & \\
2 & 10 & 0 & & & & & \\
3 & 7 & 7 & 0 & & & & \\
4 & 30 & 23 & 21 & 0 & & & \\
5 & 29 & 25 & 22 & 7 & 0 & & \\
6 & 38 & 34 & 31 & 10 & 11 & 0 & \\
7 & 42 & 36 & 36 & 13 & 17 & 9 & 0
\end{array}
$$

D(2, A) = 8.5

D(4, A) = 25.5

D(5, A) = 25.5

D(6, A) = 34.5

D(7, A) = 39.0

A = {1, 3 }

B = {2, 4, 5, 6, 7}

Divisive Clustering

$$
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & & & & & & \\
2 & 10 & 0 & & & & & \\
3 & 7 & 7 & 0 & & & & \\
4 & 30 & 23 & 21 & 0 & & & \\
5 & 29 & 25 & 22 & 7 & 0 & & \\
6 & 38 & 34 & 31 & 10 & 11 & 0 & \\
7 & 42 & 36 & 36 & 13 & 17 & 9 & 0
\end{array}
$$

D(2, A) = 8.5 D(2, B) = 29.5

D(4, A) = 25.5 D(4, B) = 13.2

D(5, A) = 25.5 D(5, B) = 15.0

D(6, A) = 34.5 D(6, B) = 16.0

D(7, A) = 39.0 D(7, B) = 18.75

A = {1, 3 }

B = {2, 4, 5, 6, 7}

Divisive Clustering

$$\begin{array}{cc} & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} & \begin{pmatrix} 0 & & & & & & \\ 10 & 0 & & & & & \\ 7 & 7 & 0 & & & & \\ 30 & 23 & 21 & 0 & & & \\ 29 & 25 & 22 & 7 & 0 & & \\ 38 & 34 & 31 & 10 & 11 & 0 & \\ 42 & 36 & 36 & 13 & 17 & 9 & 0 \end{pmatrix} \end{array}$$

D(2, A) = 8.5

D(2, B) = 29.5

Δ_2 = 21.0

D(4, A) = 25.5

D(4, B) = 13.2

Δ_4 = -12.3

D(5, A) = 25.5

D(5, B) = 15.0

Δ_5 = -10.5

D(6, A) = 34.5

D(6, B) = 16.0

Δ_6 = -18.5

D(7, A) = 39.0

D(7, B) = 18.75

Δ_7 = -20.25

A = {1, 3, 2 }

B = $\{\cancel{2}$, 4, 5, 6, 7}

Divisive Clustering

$$
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & & & & & & \\
2 & 10 & 0 & & & & & \\
3 & 7 & 7 & 0 & & & & \\
4 & 30 & 23 & 21 & 0 & & & \\
5 & 29 & 25 & 22 & 7 & 0 & & \\
6 & 38 & 34 & 31 & 10 & 11 & 0 & \\
7 & 42 & 36 & 36 & 13 & 17 & 9 & 0
\end{array}
$$

D(2, A) = 8.5	D(2, B) = 29.5	Δ_2 = 21.0
D(4, A) = 25.5	D(4, B) = 13.2	Δ_4 = -12.3
D(5, A) = 25.5	D(5, B) = 15.0	Δ_5 = -10.5
D(6, A) = 34.5	D(6, B) = 16.0	Δ_6 = -18.5
D(7, A) = 39.0	D(7, B) = 18.75	Δ_7 = -20.25

A = {1, 3, 2 }

B = { 4, 5, 6, 7}

Divisive Clustering

$$
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & & & & & & \\
2 & 10 & 0 & & & & & \\
3 & 7 & 7 & 0 & & & & \\
4 & 30 & 23 & 21 & 0 & & & \\
5 & 29 & 25 & 22 & 7 & 0 & & \\
6 & 38 & 34 & 31 & 10 & 11 & 0 & \\
7 & 42 & 36 & 36 & 13 & 17 & 9 & 0
\end{array}
$$

D(4, A) = 24.7

D(5, A) = 25.3

D(6, A) = 34.3

D(7, A) = 38.0

A = {1, 3, 2 }

B = {4, 5, 6, 7}

Divisive Clustering

$$
\begin{array}{c|c}
 & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} &
\begin{pmatrix}
0 & & & & & & \\
10 & 0 & & & & & \\
7 & 7 & 0 & & & & \\
30 & 23 & 21 & 0 & & & \\
29 & 25 & 22 & 7 & 0 & & \\
38 & 34 & 31 & 10 & 11 & 0 & \\
42 & 36 & 36 & 13 & 17 & 9 & 0
\end{pmatrix}
\end{array}
$$

D(4, A) = 24.7

D(4, B) = 10.0

D(5, A) = 25.3

D(5, B) = 11.7

D(6, A) = 34.3

D(6, B) = 10.0

D(7, A) = 38.0

D(7, B) = 13.0

A = {1, 3, 2 }

B = {4, 5, 6, 7}

Divisive Clustering

$$
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & & & & & & \\
2 & 10 & 0 & & & & & \\
3 & 7 & 7 & 0 & & & & \\
4 & 30 & 23 & 21 & 0 & & & \\
5 & 29 & 25 & 22 & 7 & 0 & & \\
6 & 38 & 34 & 31 & 10 & 11 & 0 & \\
7 & 42 & 36 & 36 & 13 & 17 & 9 & 0
\end{array}
$$

D(4, A) = 24.7	D(4, B) = 10.0	Δ_4 = -14.7
D(5, A) = 25.3	D(5, B) = 11.7	Δ_5 = -13.6
D(6, A) = 34.3	D(6, B) = 10.0	Δ_6 = -24.3
D(7, A) = 38.0	D(7, B) = 13.0	Δ_7 = -25.0

A = {1, 3, 2 }

B = {4, 5, 6, 7}

All differences are negative. The process would continue on each subgroup separately.

Spatial Clustering: Methods

K-means	Hierarchical Clustering	EM Clustering	DBSCAN

EM Clustering: Motivation

Motivation:

- Drawback of the K-means/Dendrogram
  - Each point belongs to a single cluster
  - There is no representation that a point can belong to different clusters with different probabilities
- Use probability density to associate to each point

EM Clustering: Ideas

- Assume that we know there are k clusters
- Each cluster follows a distribution (e.g., Gaussian Distribution)
  - 1D Gaussian Distribution
    - Mean μ
    - Standard deviation σ

$$p(x|\ \langle\mu, \sigma\rangle) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

EM Clustering

Since there are k clusters, we have k distributions.

- Cluster 1
  - Gaussian Distribution
    - Mean μ_1
    - Standard deviation σ_1
- Cluster 2
  - Gaussian Distribution
    - Mean μ_2
    - Standard deviation σ_2
- …
- Cluster k
  - Gaussian Distribution
    - Mean μ_k
    - Standard deviation σ_k

EM Clustering

EM Clustering Algorithm

Step 1 (Parameter Initialization)

Initialize all μ_i and σ_i

Step 2 (Expectation)

For each point x,

For each cluster i,

Calculate the probability that x belongs to cluster i

One possible implementation:

$$p(x \in C_i) = \frac{p(x|<\mu_i, \sigma_i>)}{\sum_j p(x|<\mu_j, \sigma_j>)}$$

Step 3 (Maximization)

For each cluster i,

Calculate the mean μ_i according to the probabilities that all points belong to cluster i

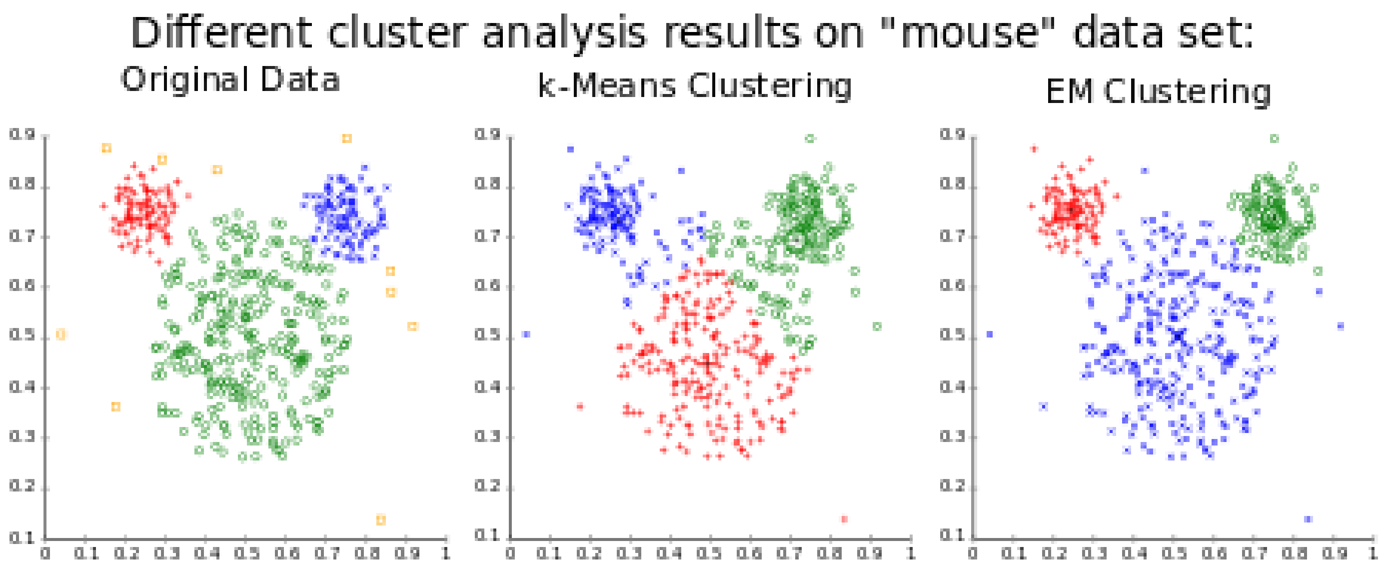
One possible implementation:

$$\mu_i = \sum_x x \cdot \frac{p(x \in C_i)}{\sum_y p(y \in C_i)}$$

Repeat Step 2 and Step 3

until the parameters conver

K-Means Clustering vs EM Clustering



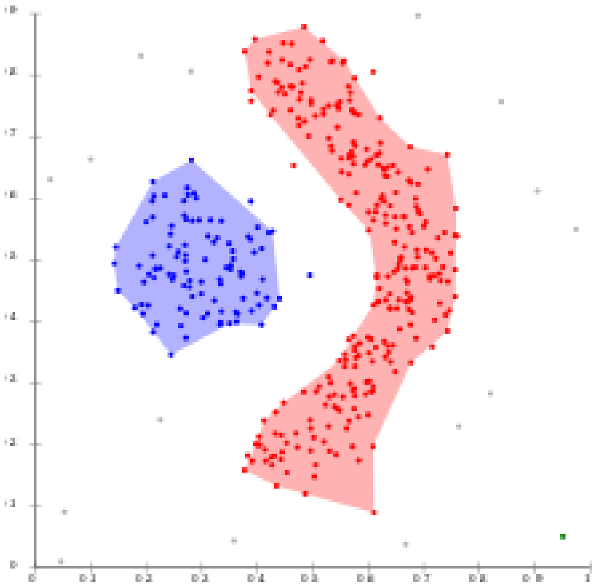
Source: Wikipedia

Spatial Clustering: Methods

K-means	Hierarchical Clustering	EM Clustering	DBSCAN

DBSCAN

- Traditional Clustering
  - Can only represent sphere clusters
  - Cannot handle irregular shaped clusters
- DBSCAN
  - **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise

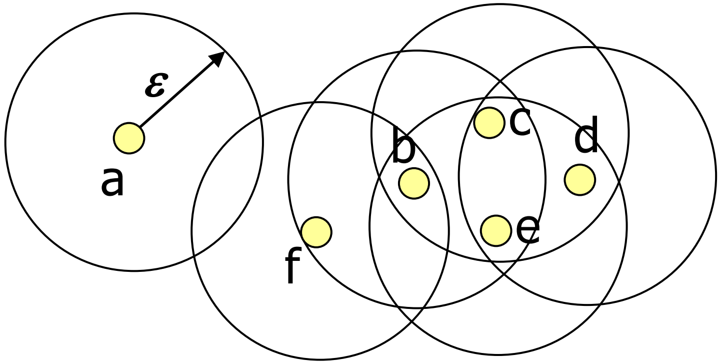


[PDF] A density-based algorithm for discovering clusters in large spatial databases with noise.
M Ester, HP Kriegel, J Sander, X Xu - Kdd, 1996 - aaai.org
Clustering algorithms are attractive for the task of class identification in spatial databases. However, the application to large spatial databases rises the following requirements for clustering algorithms: minimal requirements of domain knowledge to determine the input parameters, discovery of clusters with arbitrary shape and good efficiency on large databases. The well-known clustering algorithms offer no solution to the combination of these requirements. In this paper, we present the new clustering algorithm DBSCAN relying ...
Cited by 19006 Related articles All 77 versions

DBSCAN

Given a point p and a non-negative real number ε,

the ***ε-neighborhood*** of point p, denoted by N(p), is the set of points q (including point p itself) such that the distance between p and q is within ε.

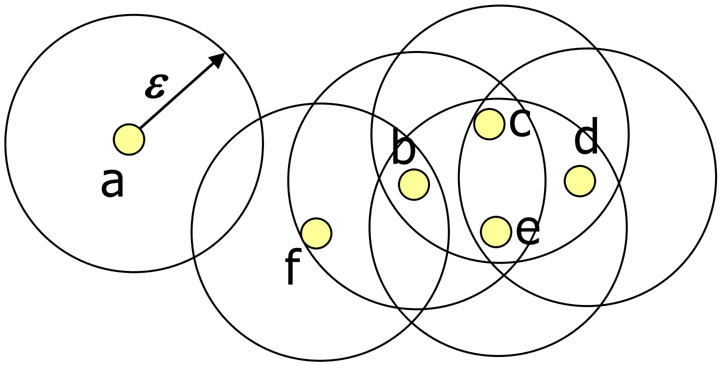


DBSCAN

Given a point p and a non-negative integer MinPts,

- **Core points:** if the size of N(p) is at least MinPts
- **Border points:** if it is not a core point but N(p) contains at least one core point.
- **Noise points:** if it is neither a core point nor a border point.

MinPts = 3



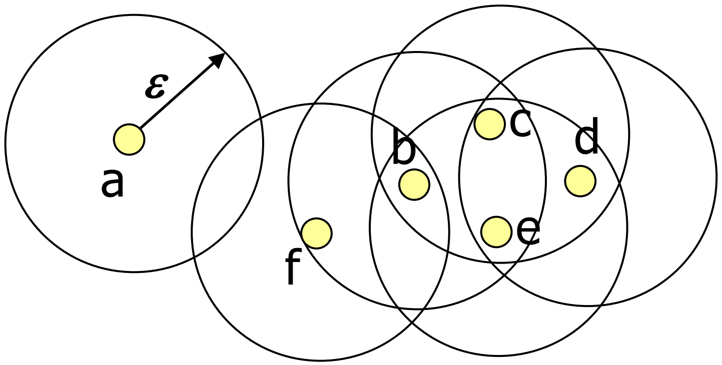
Which are core points?
Which are border points?
Which are noise points?

DBSCAN

Given a point p and a non-negative integer MinPts,

- **Core points:** if the size of N(p) is at least MinPts
- **Border points:** if it is not a core point but N(p) contains at least one core point.
- **Noise points:** if it is neither a core point nor a border point.

MinPts = 3



Core points: b, c, d, e
Border points: f
Noise points: a

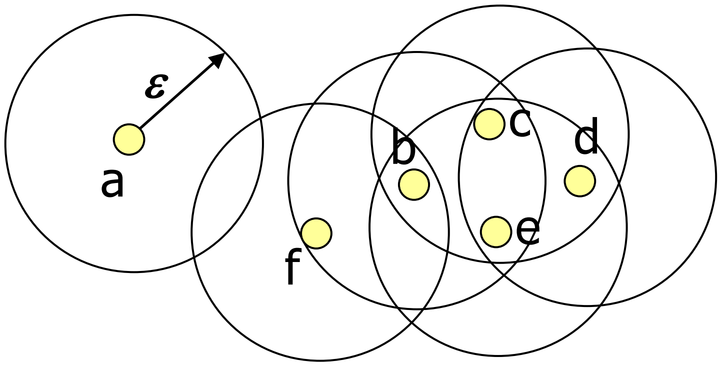
DBSCAN

- **Principle 1:** Each cluster contains at least one core point.
- **Principle 2:** Given any two core points p and q, if N(p) contains q (or N(q) contains p), then p and q are in the same cluster.
- **Principle 3:** Consider a border point p to be assigned to one of the clusters formed by Principle 1 and Principle 2. Suppose N(p) contains multiple core points. A border point p is assigned arbitrarily to one of the clusters containing these core points (formed by Principle 1 and Principle 2).
- **Principle 4:** All noise points do not belong to any clusters.

DBSCAN

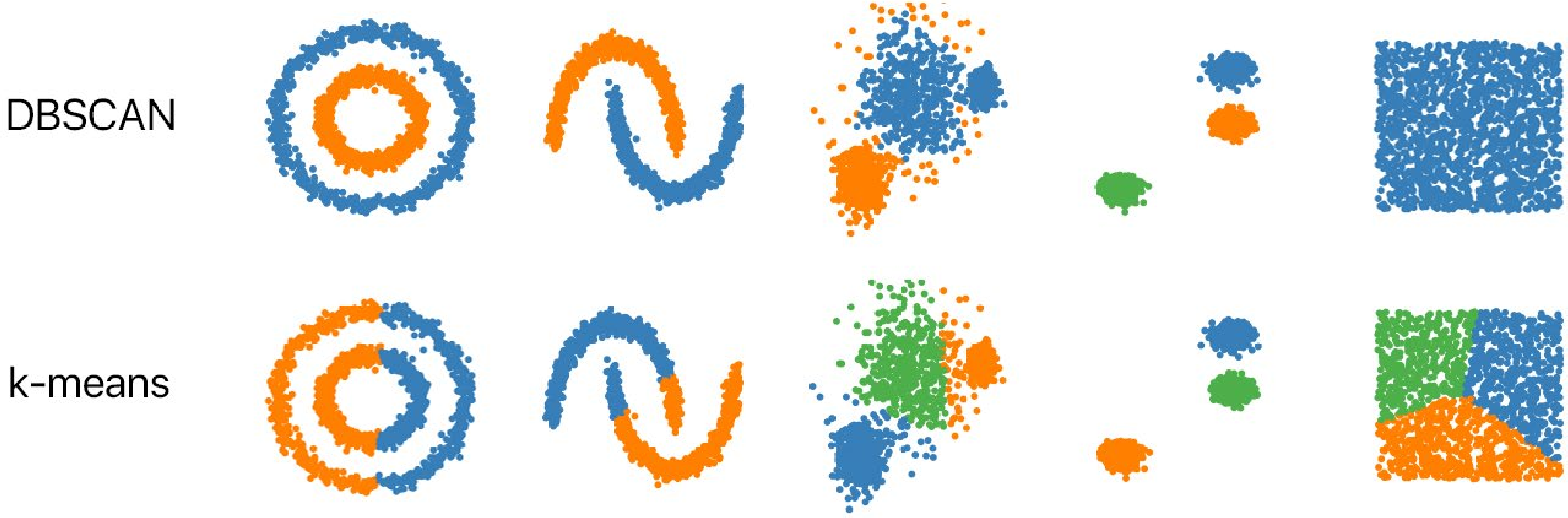
One cluster: {b, c, d, e, f}

MinPts = 3



Core points: b, c, d, e
Border points: f
Noise points: a

DBSCAN vs K-Means Clustering



https://towardsdatascience.com/understanding-dbscan-algorithm-and-implementation-from-scratch-c256289479c5

Spatial Data Mining

Spatial clustering (Hotspot)

Spatial outlier detection

Co-location mining

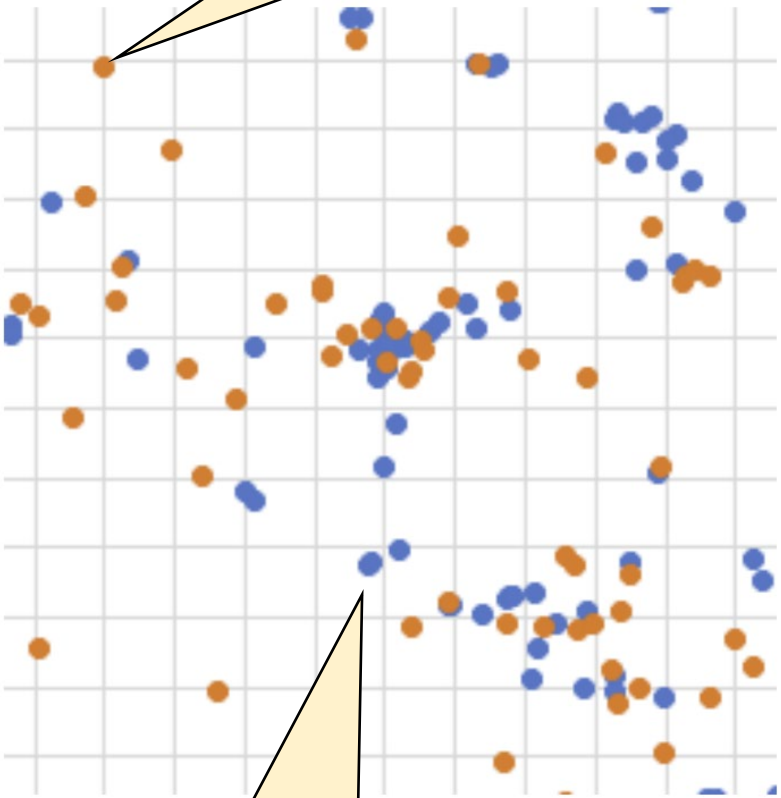
Spatial Outliers: Cases

Spatial Data (as Targets)

Spatial Data (as Contexts)

Spatial Outliers: Spatial Data (as Targets) - Example

Which POIs are outliers (i.e., the locations deviate from others)?



Orange dot: recreation

Blue dot: supermarket

Spatial Outliers: Spatial Data (as Targets) - Methods

Distance-based Model

Density-based Model

Distance-based Model

Distance-based Model (Major idea):

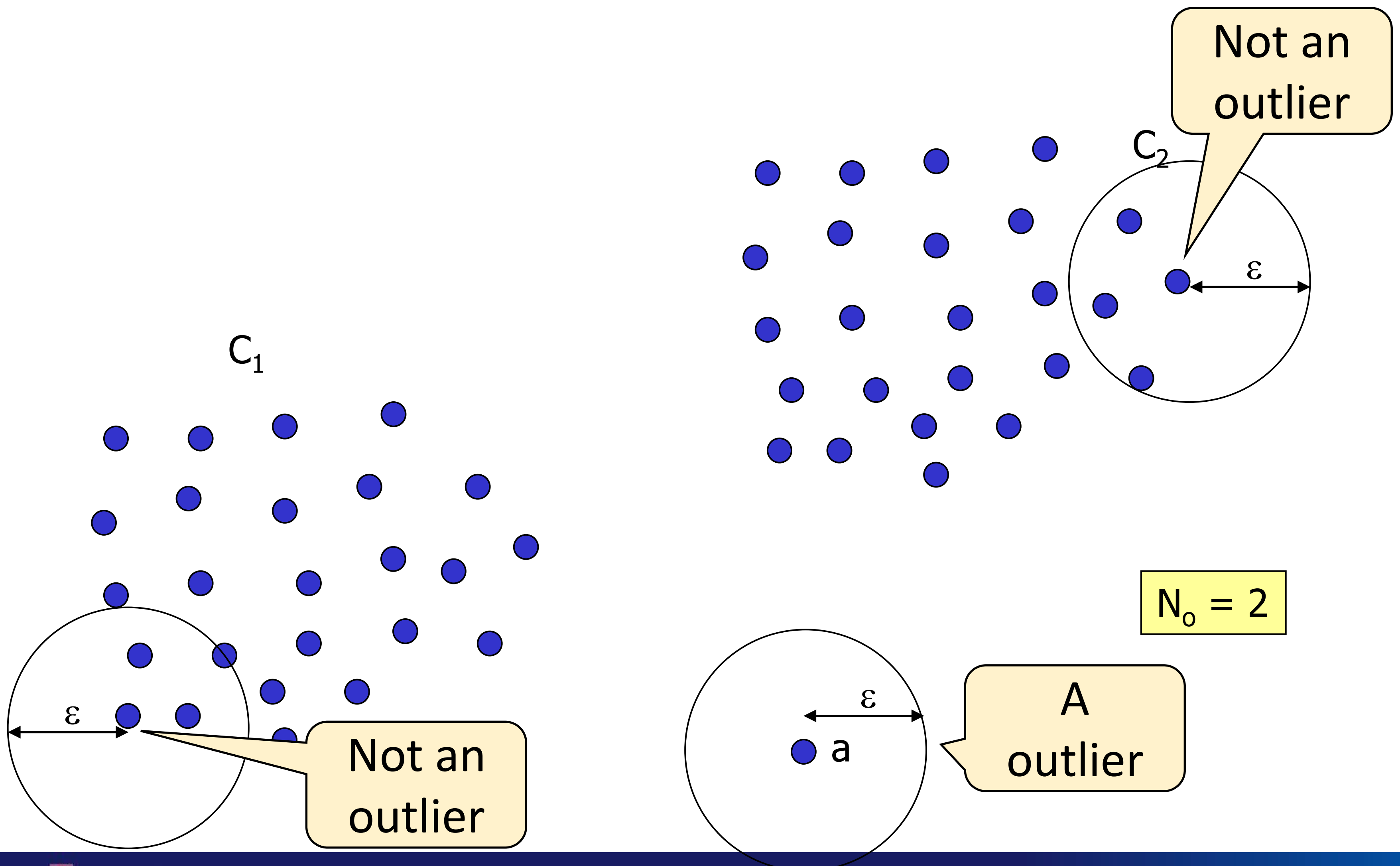
A point p is considered as an **outlier** if there are too few data points which are close to p

Distance-based Model

Distance-based Model (Definition):

1. Given a point p and a non-negative real number ε, the ***ε-neighborhood*** of point p, denoted by N(p), is the set of points q (including point p itself) such that the distance between p and q is within ε.
2. Given a non-negative integer N_o and a non-negative real number ε, a point p is said to be an **outlier** if $N(p) <= N_o$

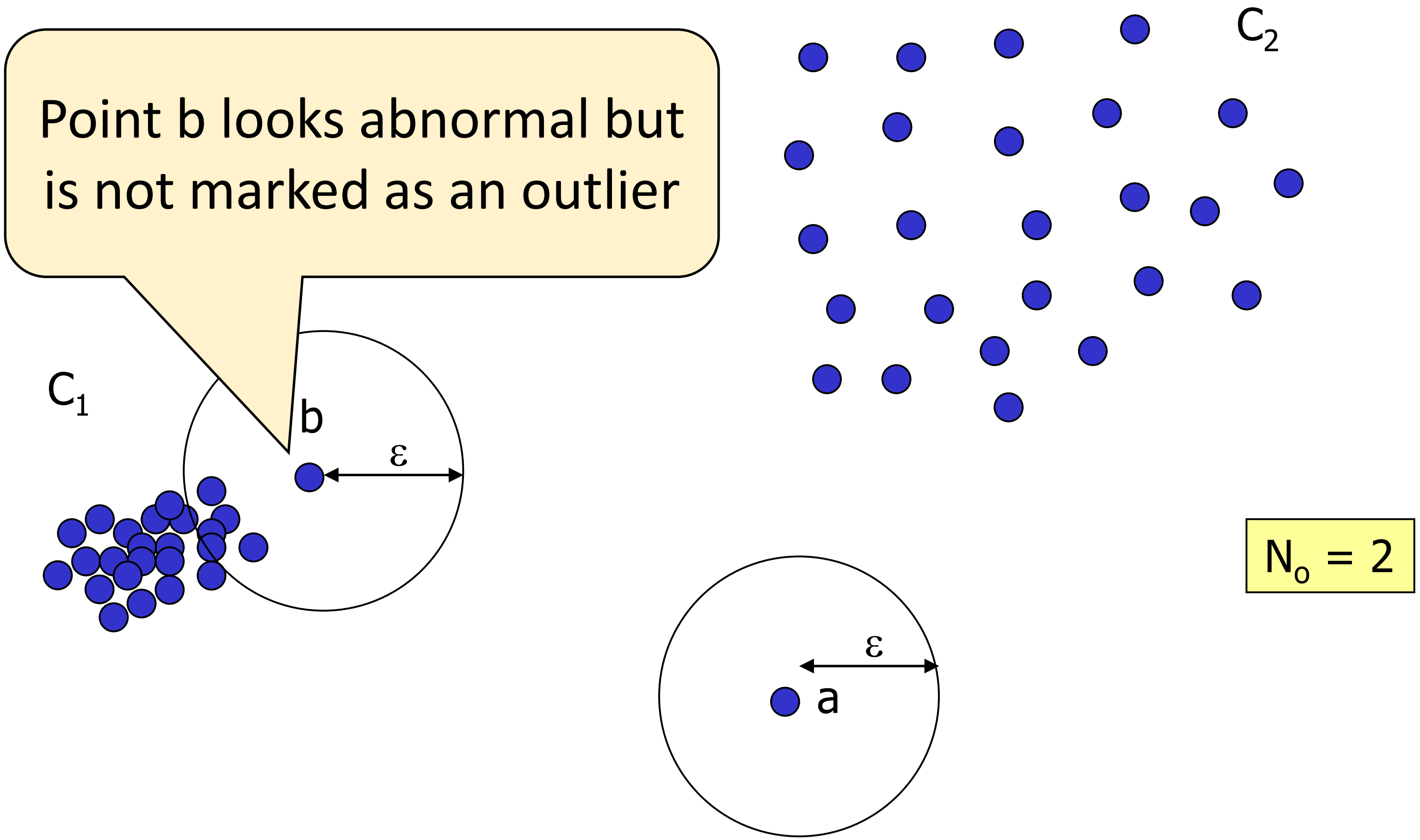
Distance-based Model



Distance-based Model

Distance-based Model may not work perfectly in some cases.

Distance-based Model

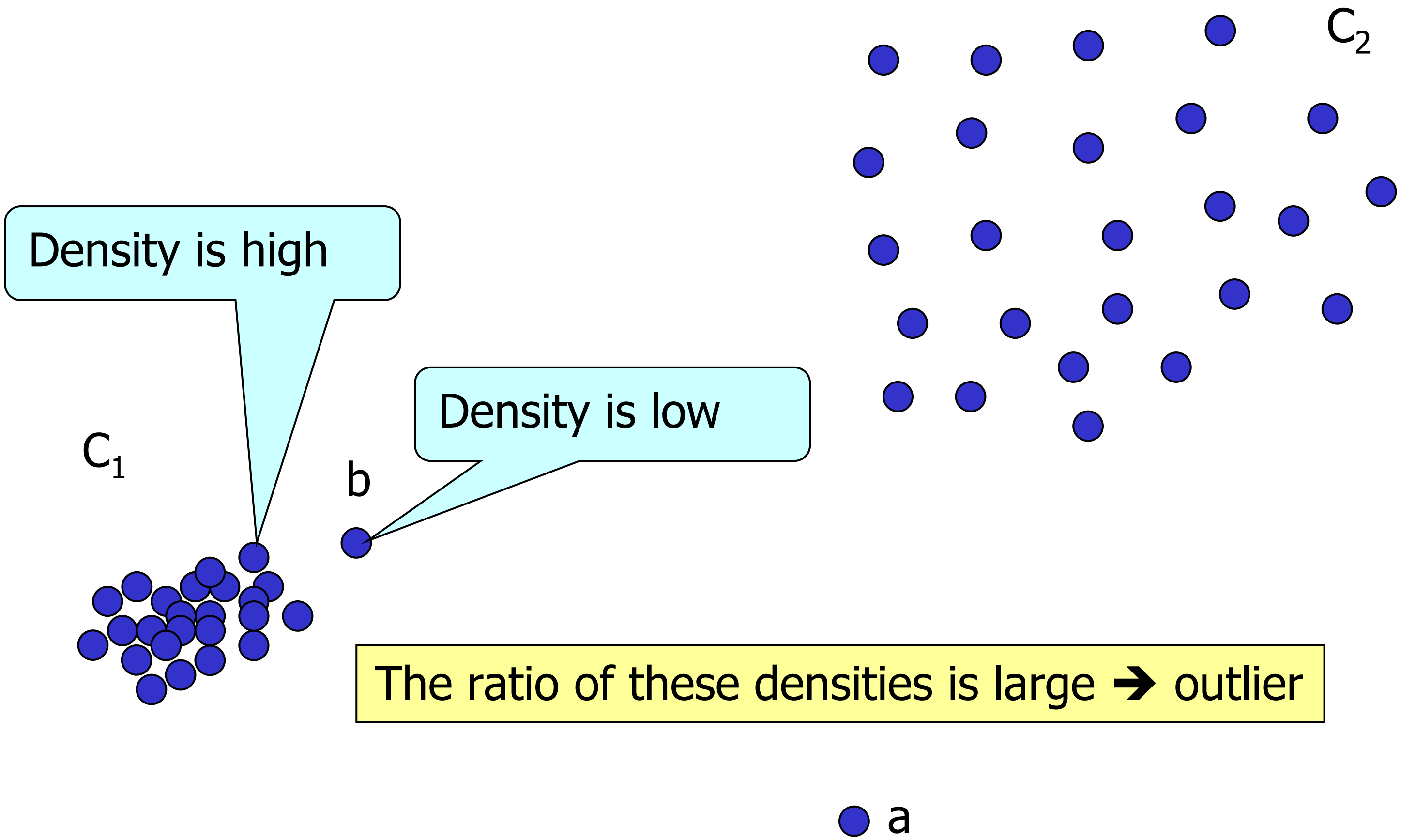


Spatial Outliers: Spatial Data (as Targets) - Methods

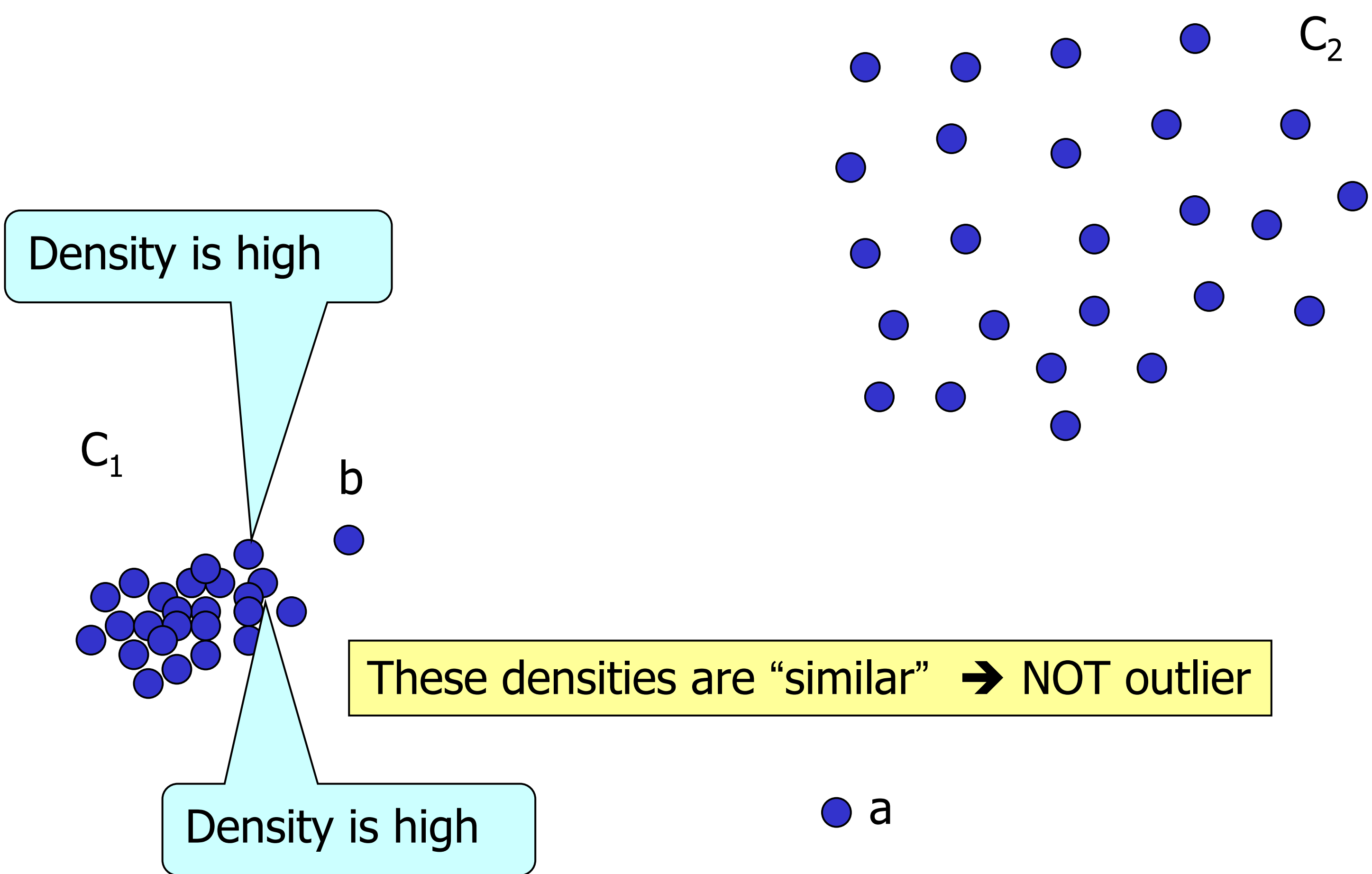
Distance-based Model

Density-based Model

Density-based Model: Main Idea



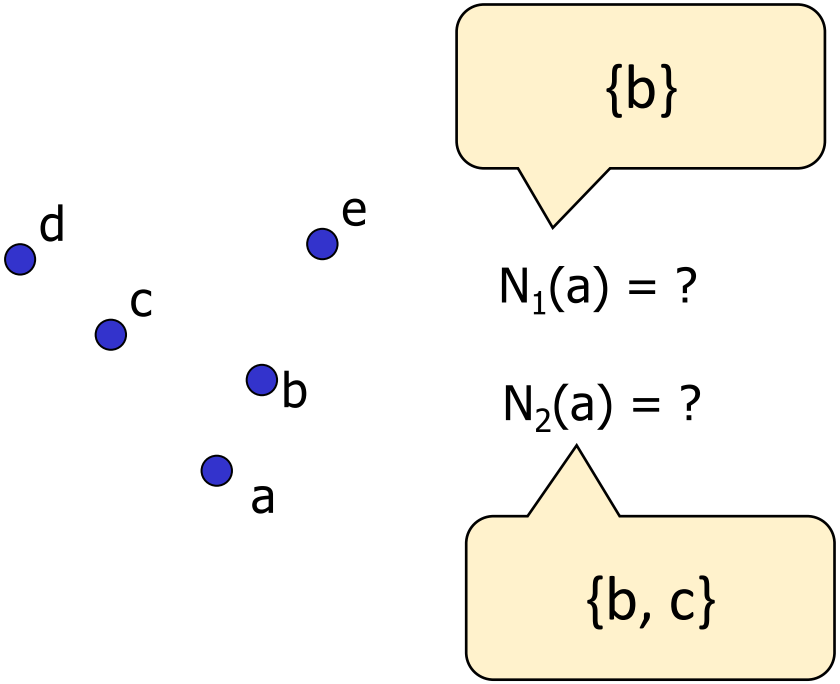
Density-based Model: Main Idea



Density-based Model

Given an integer k and a point p,

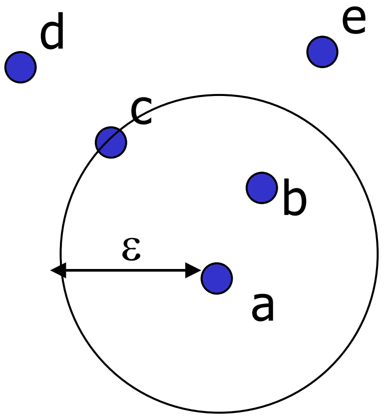
- $N_k(p)$ is defined to be the ε-neighborhood of p (excluding point p)
- where ε is the distance between p and the k-th nearest neighbor



Density-based Model

Reachability Distance of p with respect to o: Given two points p and o and an integer k,

- **Reach_dist$_k$(p, o)** is defined to be max{dist(p, o), ε}
- where ε is the distance between p and the k-th nearest neighbor



k = 2

Reach_dist$_2$(a, b) =? ε

Reach_dist$_2$(a, c) =? ε

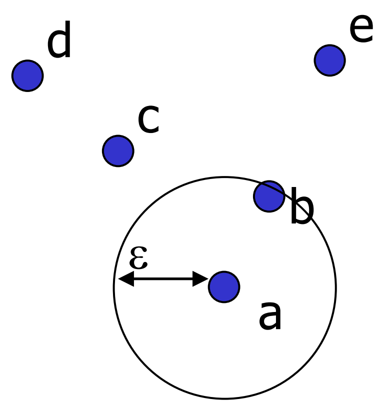
Reach_dist$_2$(a, d) =? dist(a, d)

Reach_dist$_2$(a, e) =? dist(a, e)

Density-based Model

The **local reachability density** of p (denoted by **$\mathbf{lrd_k(p)}$**) is defined to be $1/\varepsilon$

- where ε is the distance between p and the k-th nearest neighbor



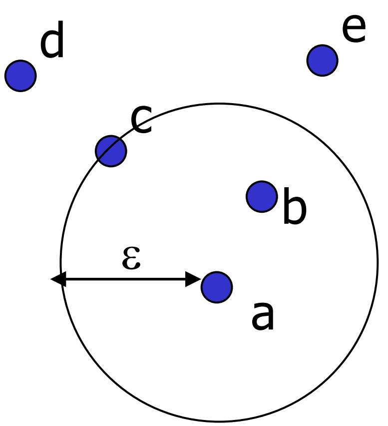
k = 1

lrd_1 (a) = 1/dist(a, b)

Density-based Model

The **local reachability density** of p (denoted by $\mathbf{lrd_k(p)}$) is defined to be $1/\varepsilon$

- where ε is the distance between p and the k-th nearest neighbor



k = 2

$lrd_2(a) = 1/dist(a, c)$

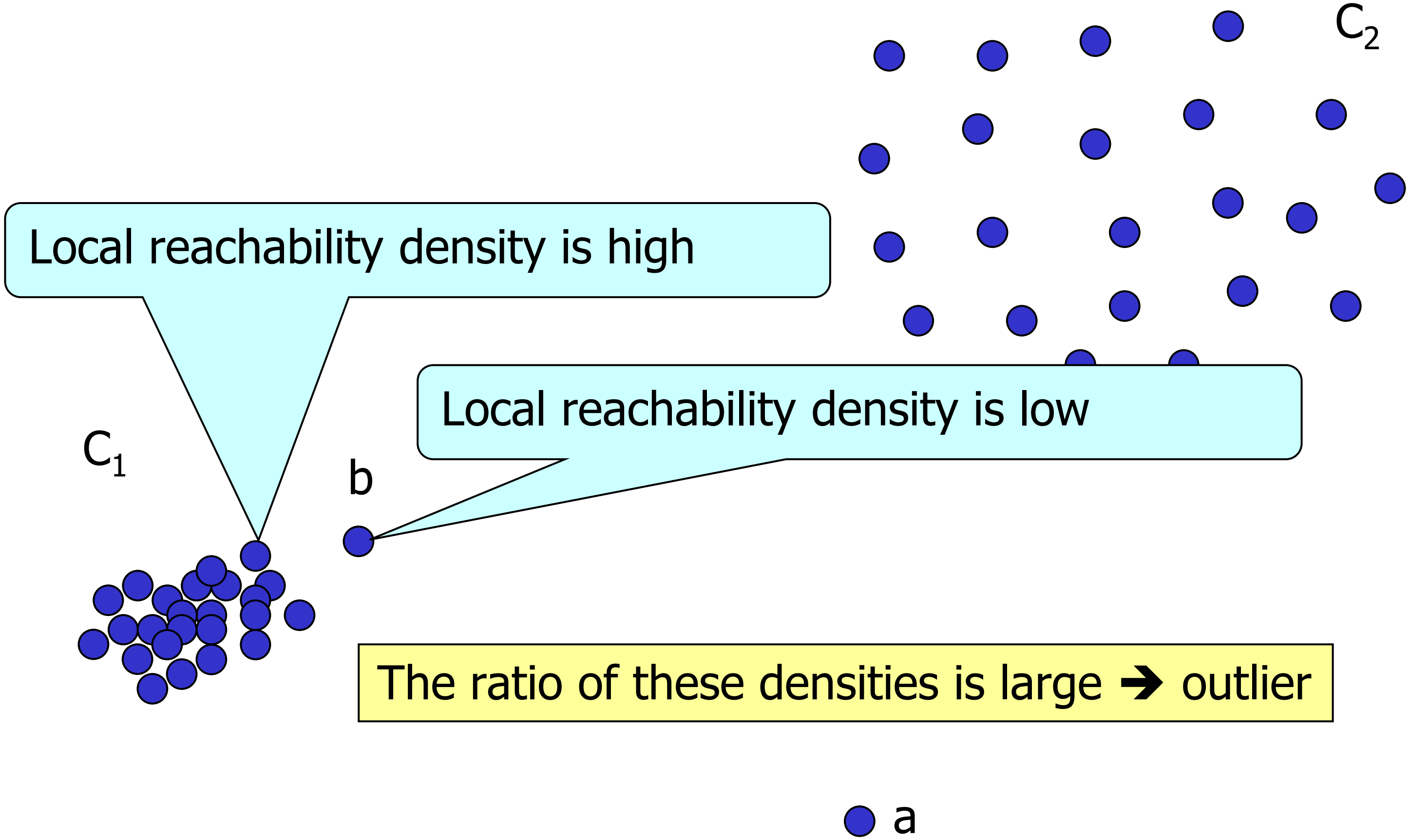
Density-based Model

The **local outlier factor (LOF)** of a point p is equal to

$$\frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{k}$$

The outlier factor is higher if the ratio is higher

Density-based Model



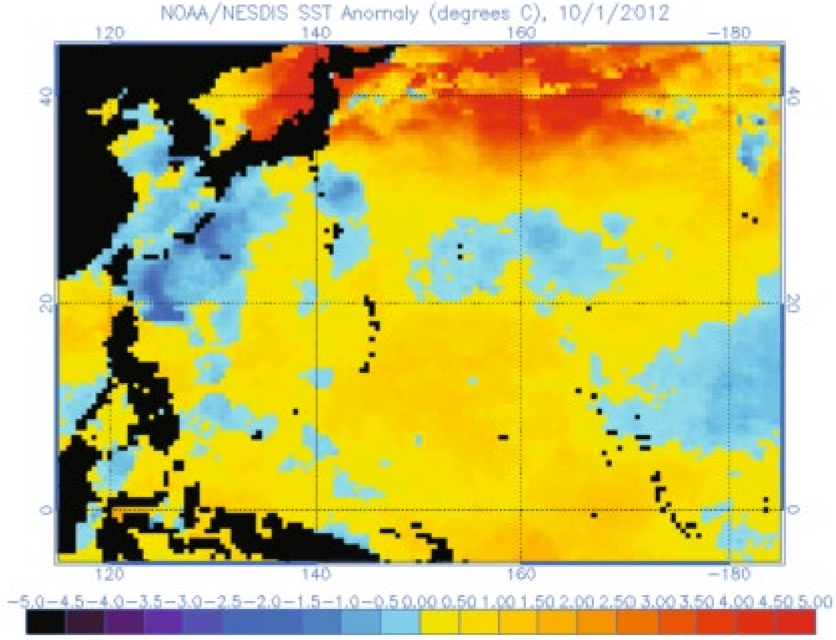
Spatial Outliers: Cases

Spatial Data (as Targets)

Spatial Data (as Contexts)

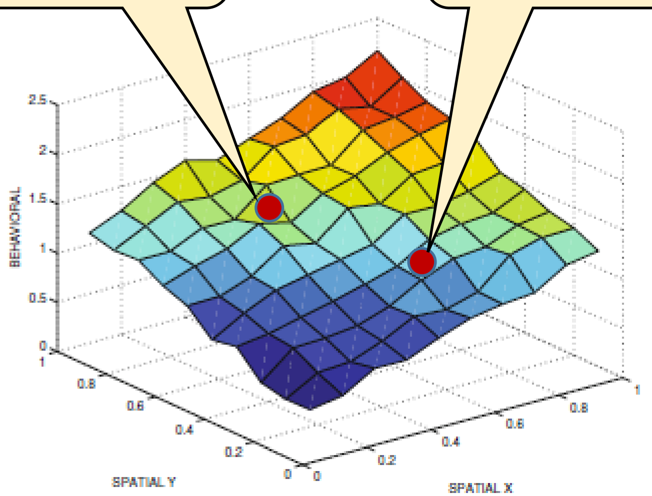
Spatial Outliers: Spatial Data (as Contexts) - Example

Contexts: spatial data
Behaviors: temperatures



Sea surface temperature anomalies
[source: NOAA Satellite and Information Service]

Spatial Data as Contexts: Method (1) – Variogram Cloud

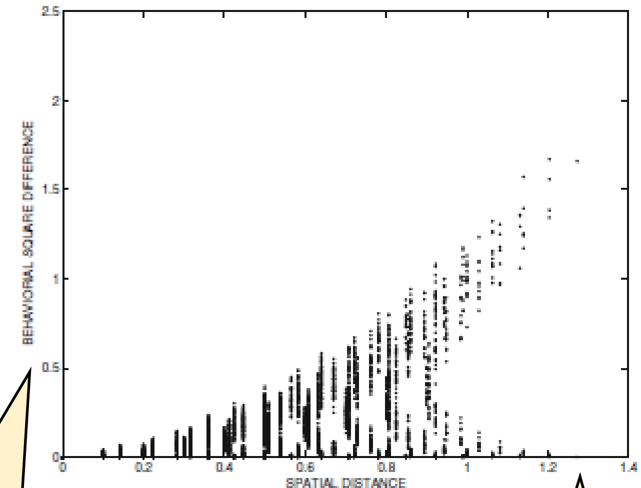


(a) Smooth spatial variation with no outlier

Spatial distance (a, b)

Behavior difference square (a. b)

The behavior difference square is positively correlated with the distance



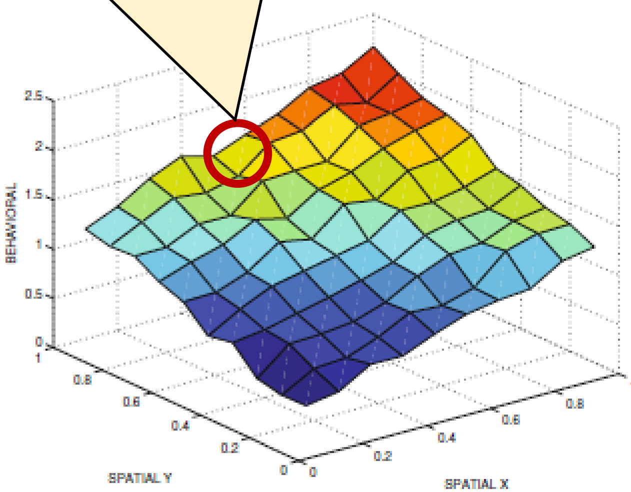
(b) Variogram Cloud with no outlier

Behavior difference square

Spatial distance

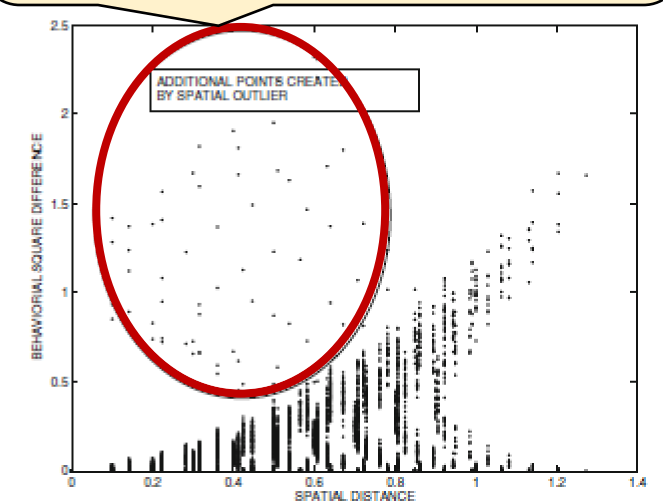
Spatial Data as Contexts: Method (1) – Variogram Cloud

Suppose we add some **outliers** here (not visible)



(a) Smooth spatial variation with no outlier

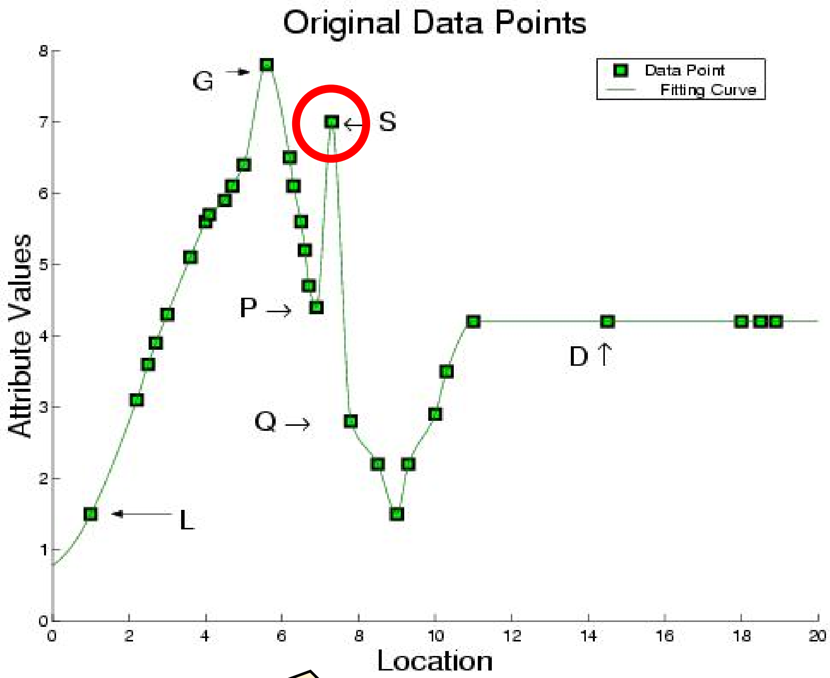
Additional points creates by **spatial outliers**



(d) Variogram Cloud with added outlier

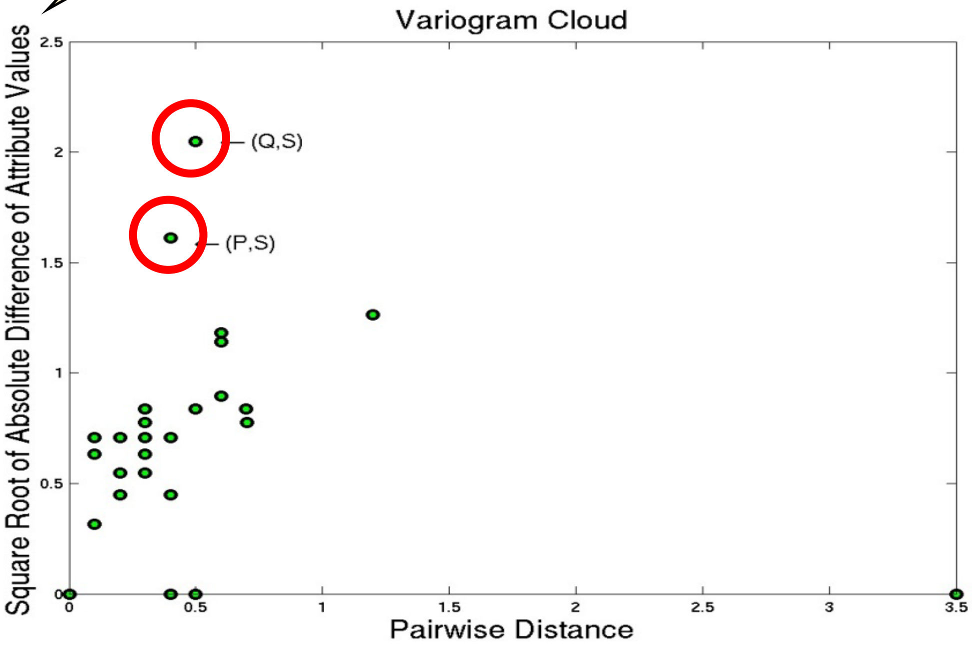
The outliers can be traced out as indicated by the points in the circle

Spatial Data as Contexts: Method (1) – Variogram Cloud



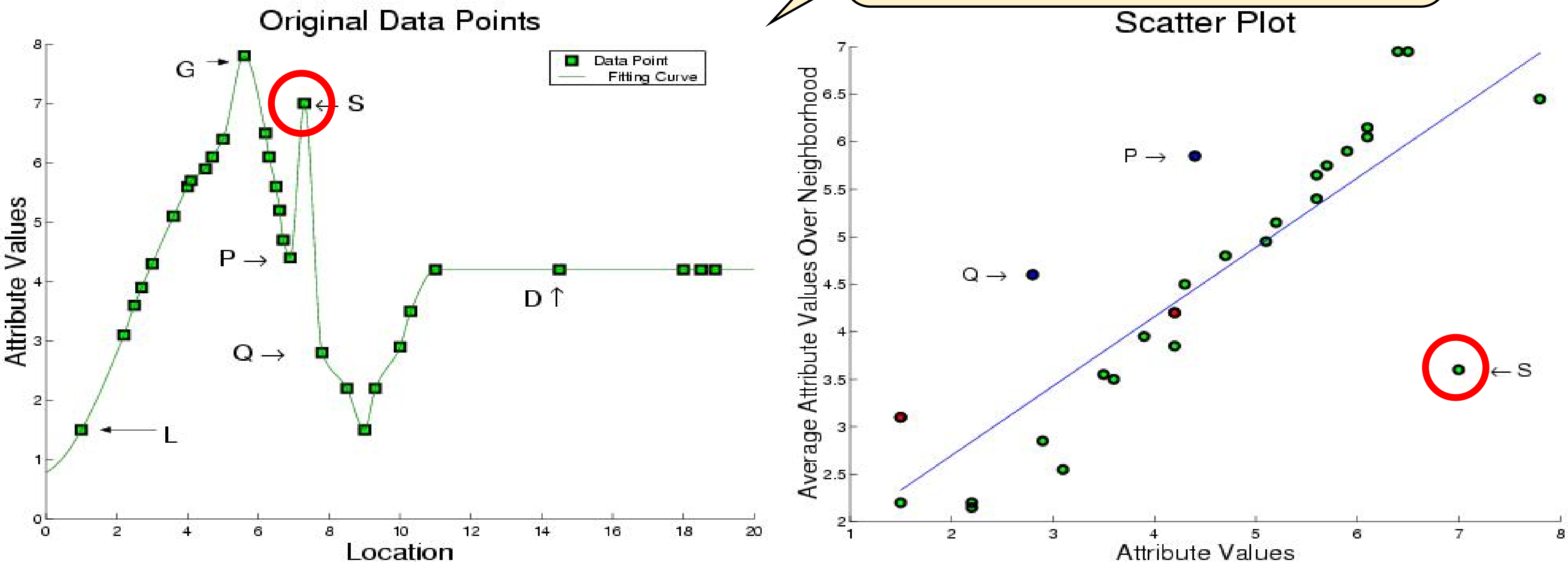
One-dimensional space

Square Root of Absolute Difference



Pairwise distance

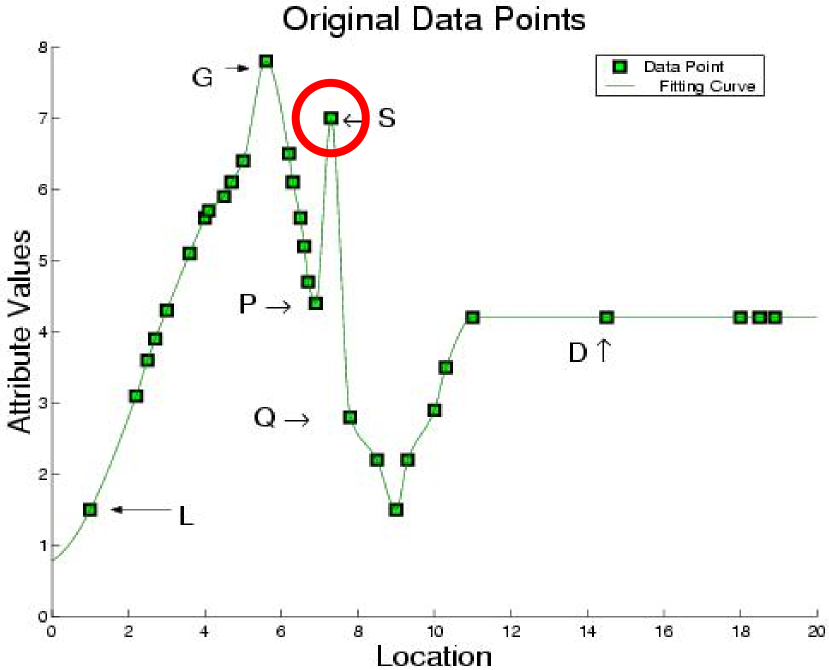
Spatial Data as Contexts: Method (2) – Scatter Plot

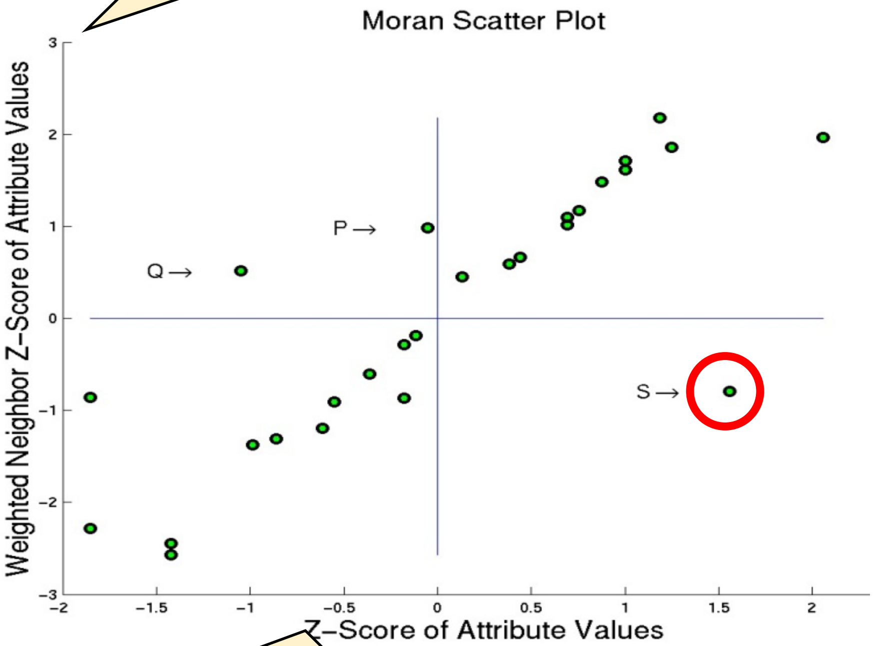


Average Attribute Values over Neighborhood

Attribute Values

Spatial Data as Contexts: Method (3) – Moran Scatterplot





Weighted Neighbor Z-score

Z-score

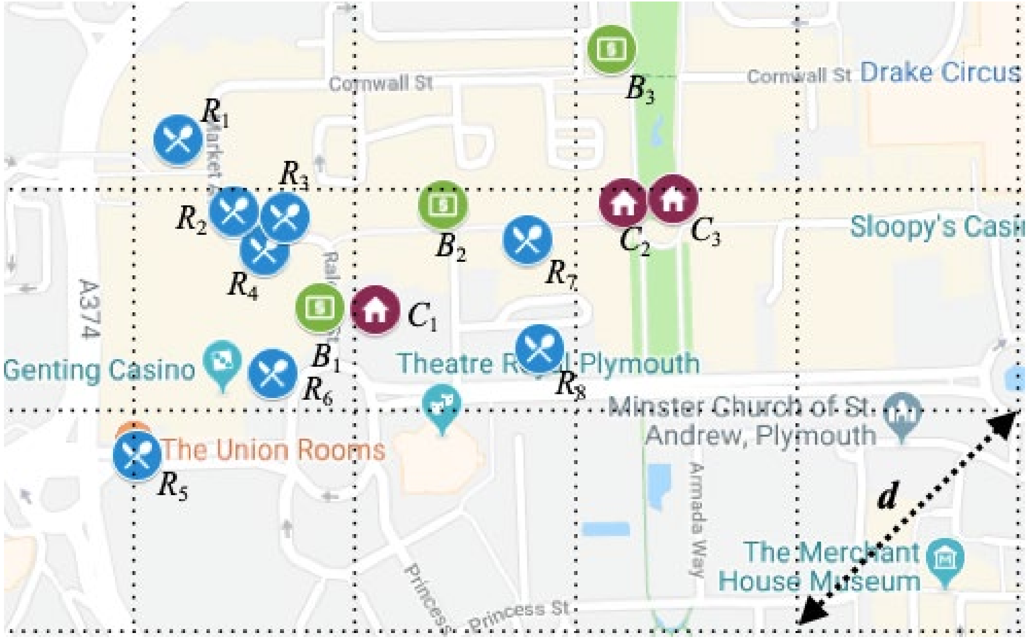
$$z = (x - \mu) / \sigma$$

Spatial Data Mining

Spatial clustering (Hotspot)	**Spatial outlier detection**	**Co-location mining**

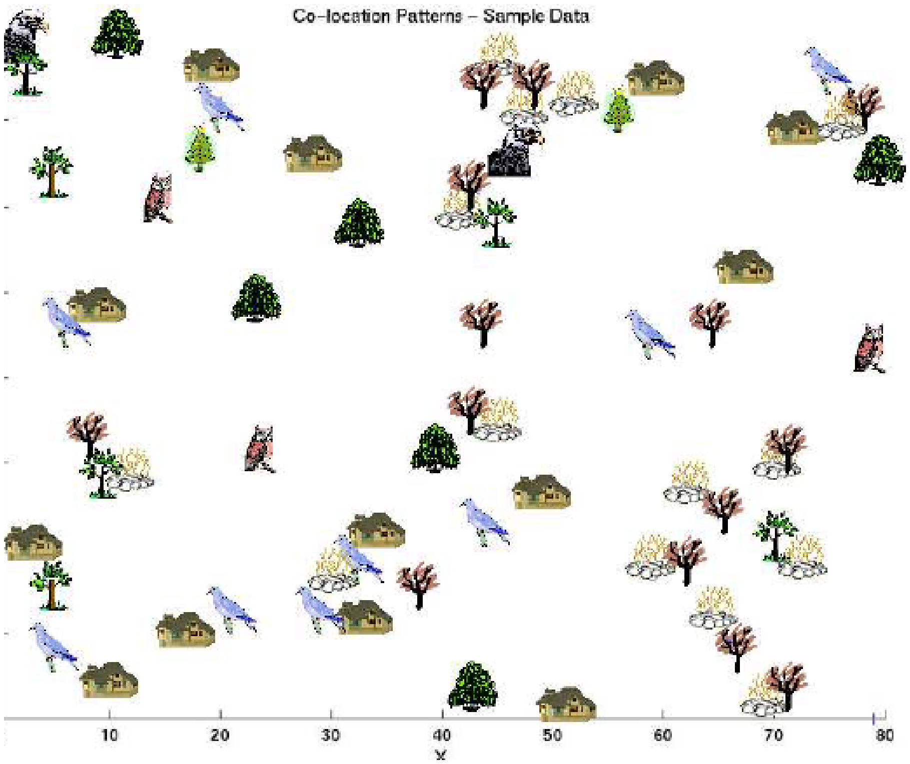
Co-location Mining: Examples

What are POI types that are **located nearby (co-located)** often?



Co-location Mining: Examples

What are species that are **located nearby (co-located)** often?



Source: Discovering Spatial Co-location Patterns: A General Approach, IEEE Transactions on Knowledge and Data Eng., 16(12),

Co-location Mining: Background – Frequent Item Mining

Supermarket Application

Item

History or **Transaction**

Raymond: apple, coke, coffee

David: diaper, coke, …

Emily: milk, biscuit, …

Derek: coke, milk, …

An interesting association:

Diaper and **Beer** are usually bought together.

diaper beer

Co-location Mining: Background – Frequent Item Mining

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

A, D

A, B, D, E

B, C

A, B, C, D, E

B, C, E

Single Items (or simply items): A B C D E

Itemsets: {B, C} {A, B, C} {B, C, D} {A}

2-itemset 3-itemset 3-itemset 1-itemset

Co-location Mining:

Frequent itemsets:
itemsets with support >= a threshold (e.g., 3)

e.g., {A}, {B}, {B, C}
but NOT {A, B, C}

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

Support = 3

Support = 4

Single Items (or simply items): A B C D E

Itemsets: {B, C} {A, B, C} {B, C, D} {A}

1-frequent itemset of size 3

3-frequent itemset of size 2

Support = 3

Support = 1

Co-location Mining: Background – Apriori Algorithm

Suppose we want to find all "frequent" itemsets (e.g., itemsets with support >= 3)

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

{B, C} is frequent

Support of {B, C} = 3

Is {B} frequent?

Is {C} frequent?

Property 1: If an itemset S is frequent, then any proper subset of S must be frequent.

Co-location Mining: Background – Apriori Algorithm

Suppose we want to find all “frequent” itemsets (e.g., itemsets with support >= 3)

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

{B, C, E} is NOT frequent

Support of {B, C, E} = 2

Is {A, B, C, E} frequent?

Is {B, C, D, E} frequent?

Property 2: If an itemset S is NOT frequent, then any proper superset of S must NOT be frequent.

Co-location Mining: Background – Apriori Algorithm

Property 1: If an itemset S is frequent, then any proper subset of S must be frequent.

Property 2: If an itemset S is NOT frequent, then any proper superset of S must NOT be frequent.

Anti-monotonicity Property

Co-location Mining: Background – Apriori Algorithm

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

Item	Count
A	3
B	
C	
D	
E	

Co-location Mining: Background – Apriori Algorithm

Suppose we want to find all "frequent" itemsets (e.g., itemsets with support >= 3)

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

Item	Count
A	3
B	4
C	3
D	3
E	3

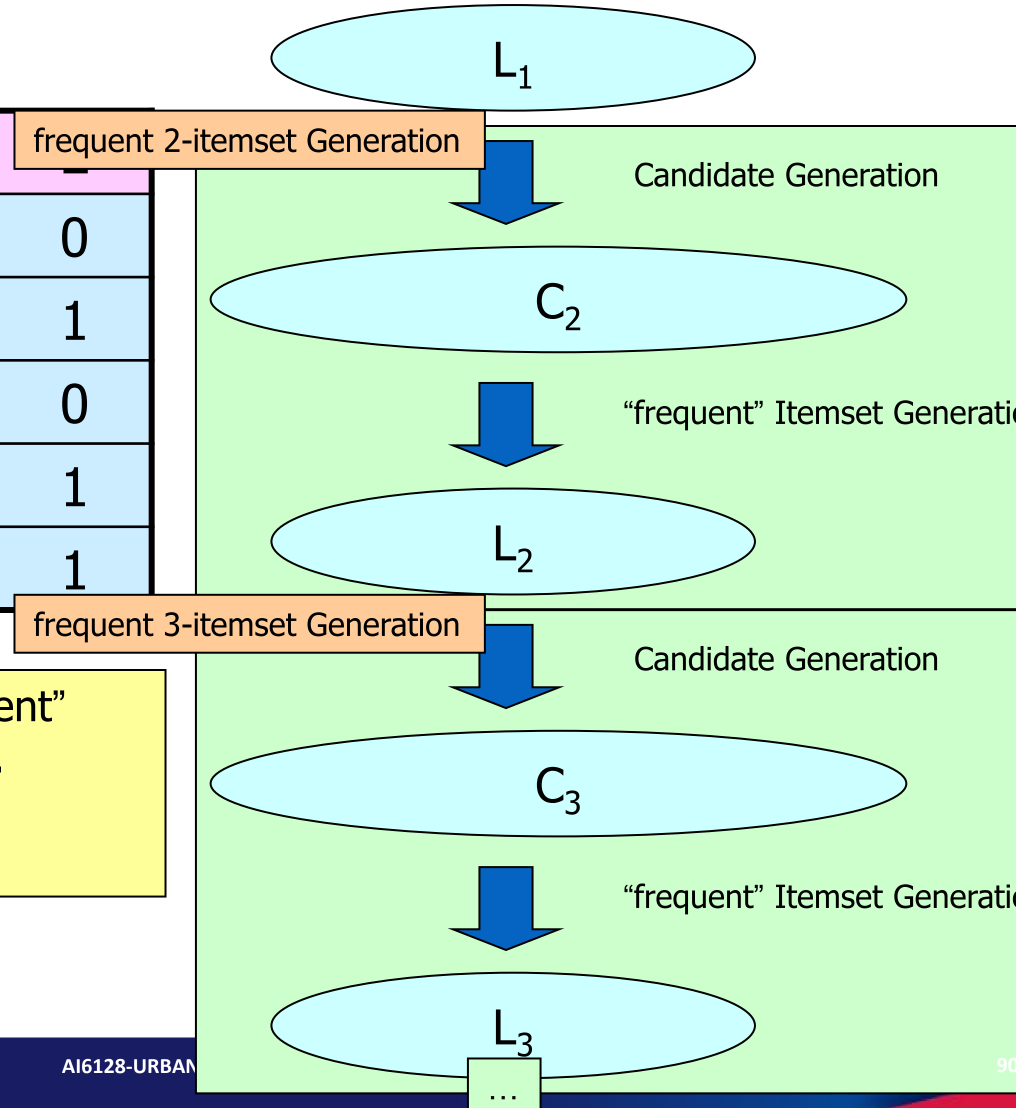
Thus, {A}, {B}, {C}, {D} and {E} are "frequent" itemsets of size 1 (or, "frequent" 1-itemsets).

We set L_1 = {{A}, {B}, {C}, {D}, {E}}

Co-location Mining: Background – Apriori Algorithm

Suppose we want to find all "frequent" itemsets (e.g., itemsets with support >= 3)

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1



Thus, {A}, {B}, {C}, {D} and {E} are "frequent" itemsets of size 1 (or, "frequent" 1-itemsets).

We set L_1 = {{A}, {B}, {C}, {D}, {E}}

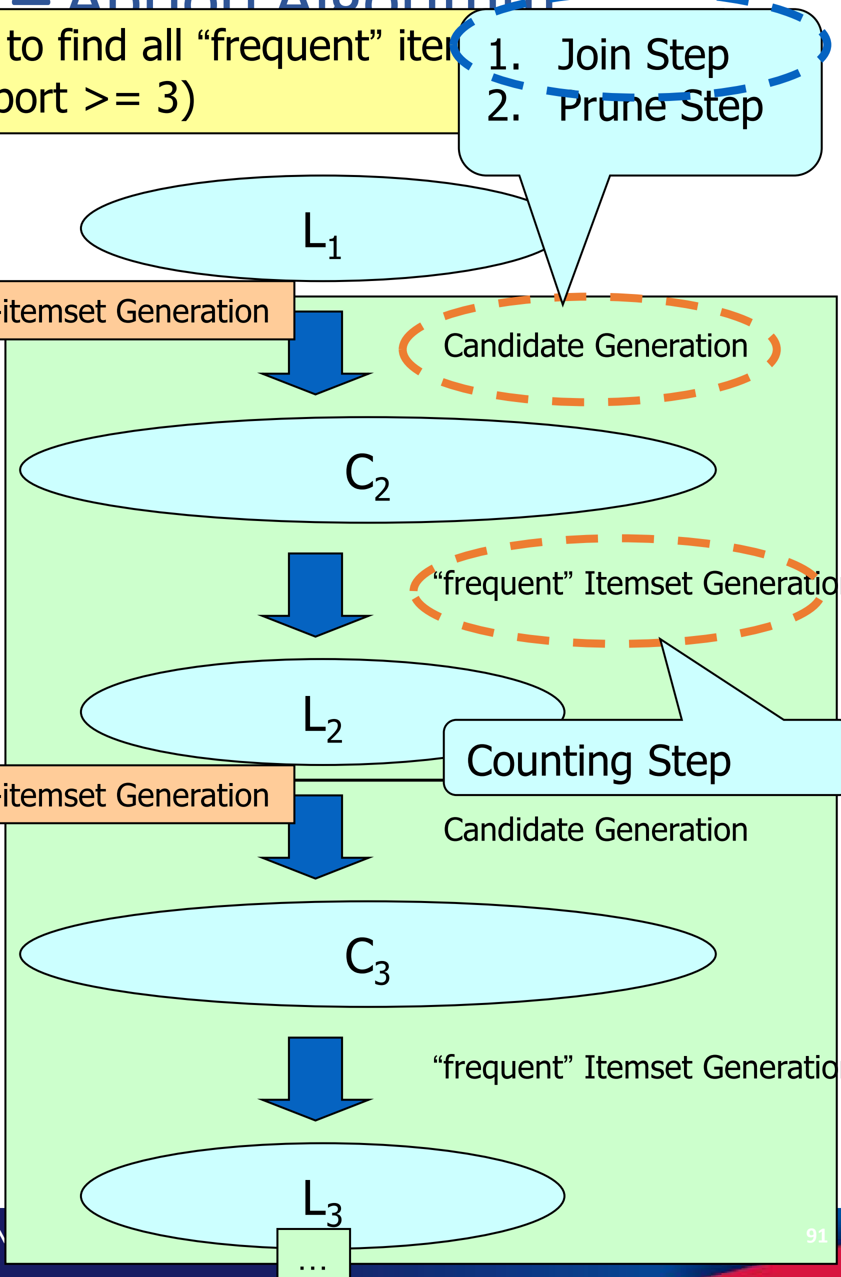
Co-location Mining: Background – Apriori Algorithm

Suppose we want to find all “frequent” ite
itemsets with support >= 3)

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

Thus, {A}, {B}, {C}, {D} and {E} are “frequent” itemsets of size 1 (or, “frequent” 1-itemsets).

We set L_1 = {{A}, {B}, {C}, {D}, {E}}



Co-location Mining: Ba

Property 1: If an itemset S is frequent, then any proper subset of S must be frequent.

Property 2: If an itemset S is NOT frequent, then any proper superset of S must NOT be frequent.

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

Suppose we know that itemset {B, C} and itemset {B, E} are frequent (i.e., L_2).

It is possible that itemset {B, C, E} is also frequent (i.e., C_3).

Co-location Mining: Background – Apriori Algorithm

Join Step

- Input: L_{k-1}, a set of all frequent (k-1)-itemsets
- Output: C_k, a set of candidates k-itemsets
- **For each** pair of p and q in L_{k-1}

  where $p.item_1 = q.item_1$,

  $p.item_2 = q.item_2$,

  …

  $p.item_{k-2} = q.item_{k-2}$,

  $p.item_{k-1} < q.item_{k-1}$

  insert into C_k the itemset $\{p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}\}$

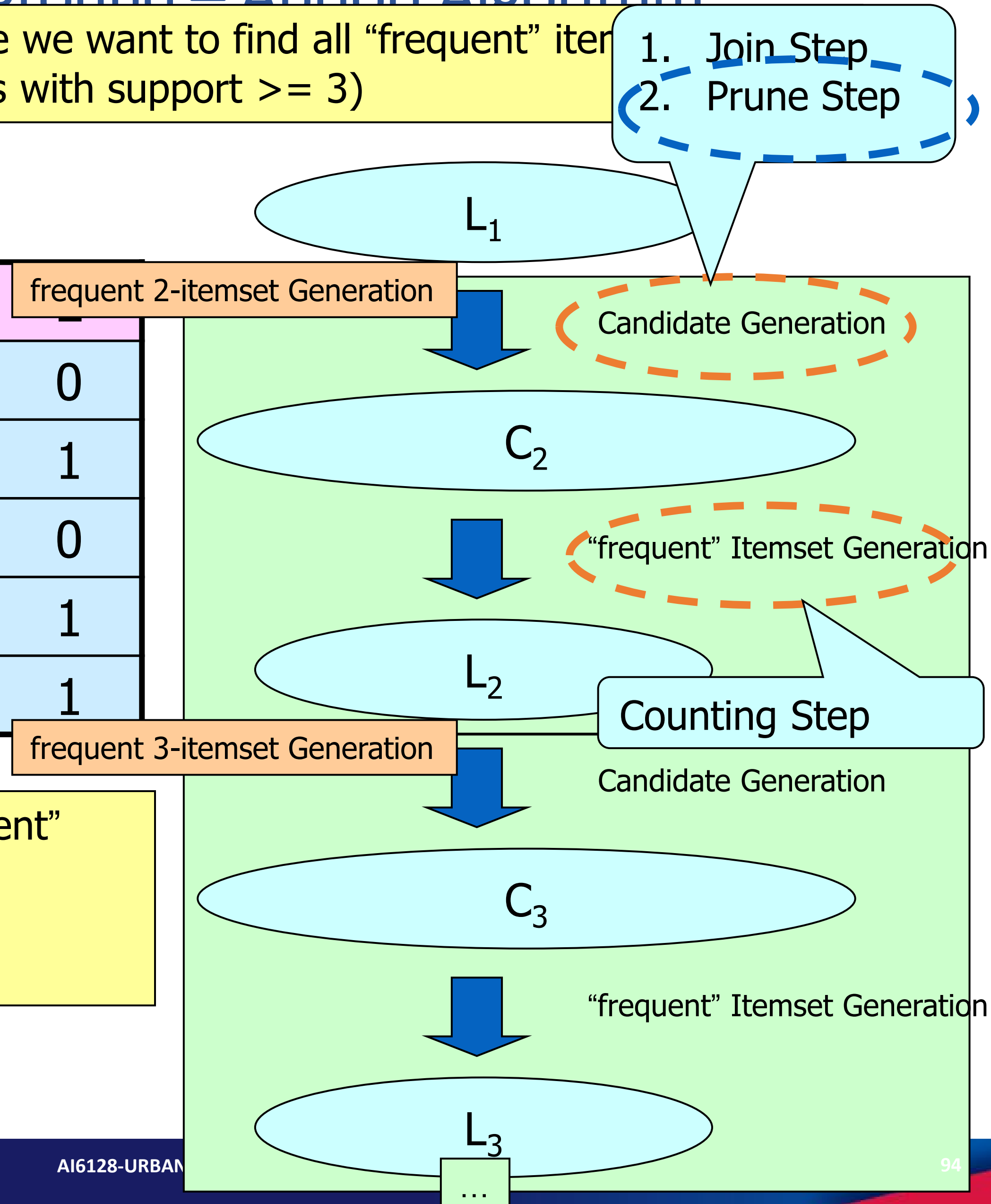
Co-location Mining: Background – Apriori Algorithm

Suppose we want to find all “frequent” iter
itemsets with support >= 3)

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

Thus, {A}, {B}, {C}, {D} and {E} are “frequent” itemsets of size 1 (or, “frequent” 1-itemsets).

We set L_1 = {{A}, {B}, {C}, {D}, {E}}



Co-location Mining: Ba

Property 1: If an itemset S is frequent, then any proper subset of S must be frequent.

Property 2: If an itemset S is NOT frequent, then any proper superset of S must NOT be frequent.

TID	A	B	C	D	E
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1

Suppose we know that itemset {B, C} and itemset {B, E} are frequent (i.e., L_2).

It is possible that itemset {B, C, E} is also frequent (i.e., C_3).

Suppose we know that {C, E} is not frequent.
We can prune {B, C, E} in C_3.

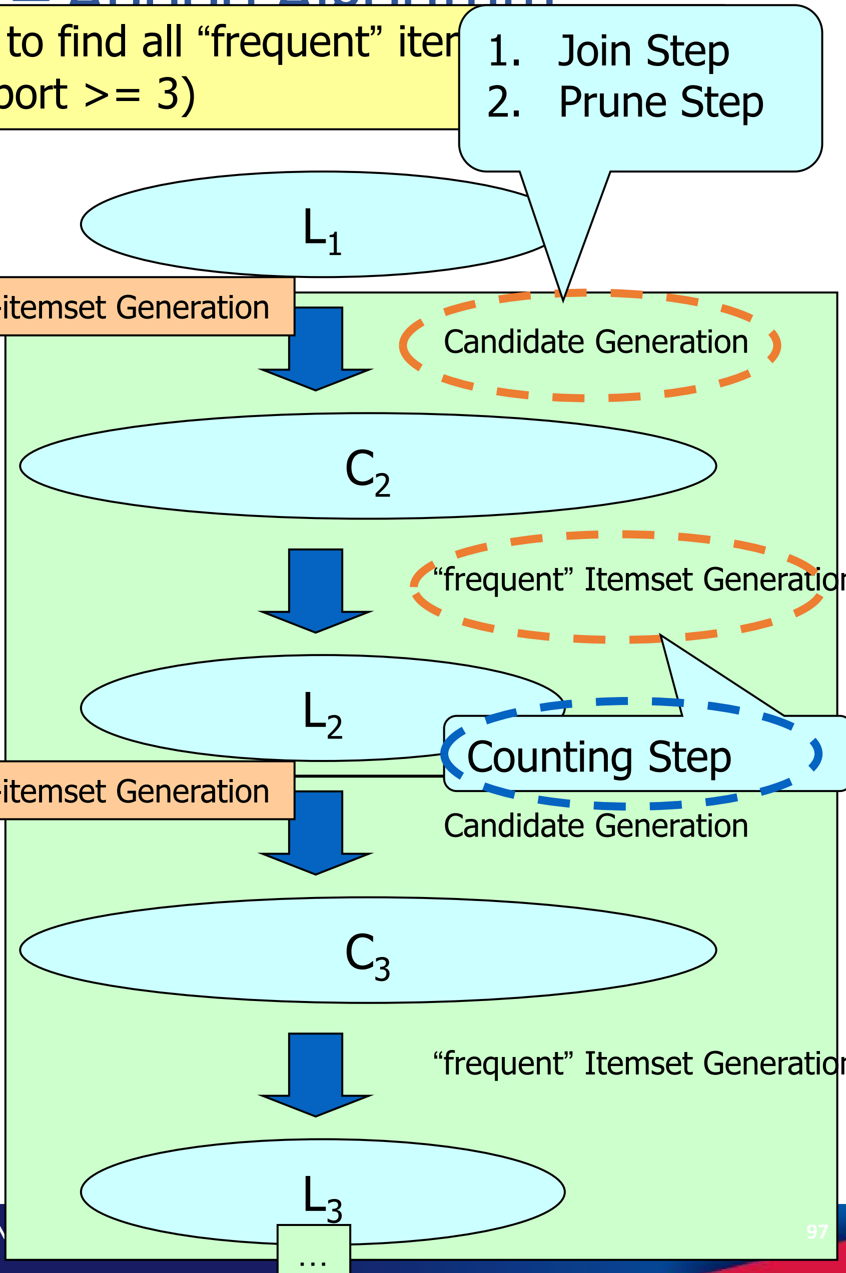
Co-location Mining: Background – Apriori Algorithm

- **Prune Step**
  - forall itemsets $c \in C_k$ (from Join Step) do
    - for all (k-1)-subsets s of c do
      - if (s not in L_{k-1}) then
        - delete c from C_k

Co-location Mining: Background – Apriori Algorithm

Suppose we want to find all “frequent” iter
itemsets with support >= 3)

TID	A	B	C	D	
t1	1	0	0	1	0
t2	1	1	0	1	1
t3	0	1	1	0	0
t4	1	1	1	1	1
t5	0	1	1	0	1



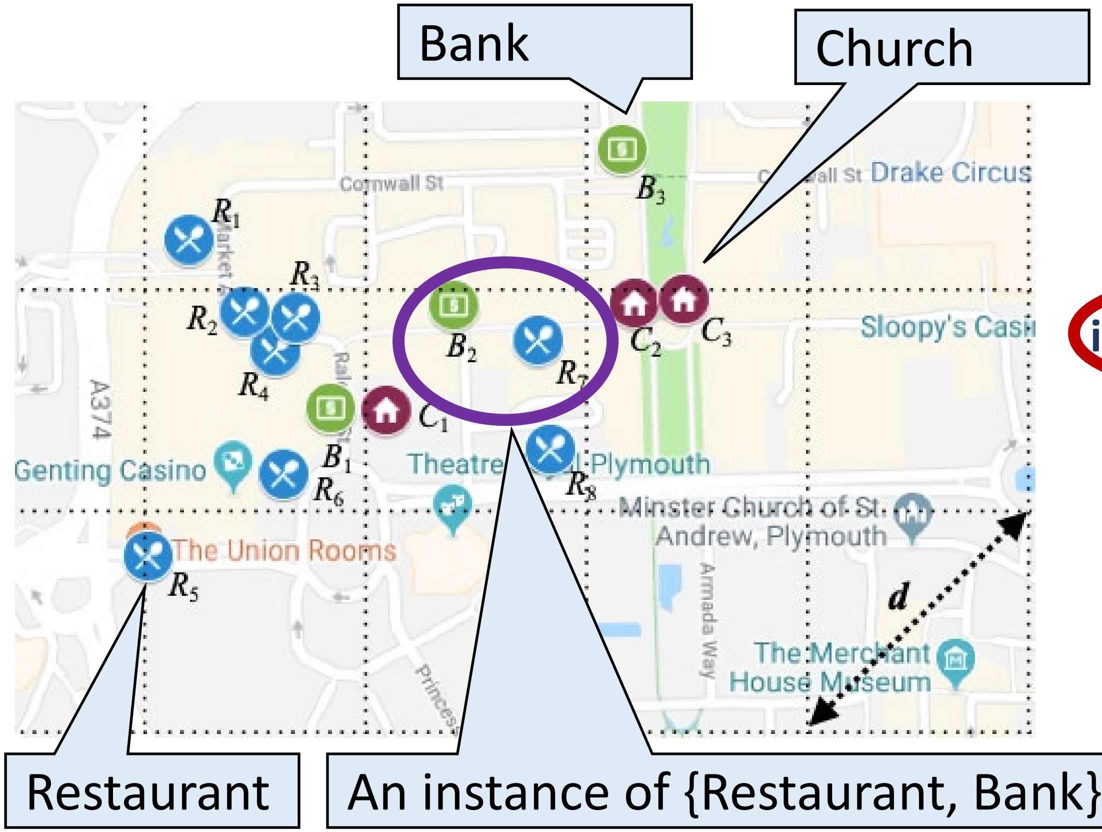
Thus, {A}, {B}, {C}, {D} and {E} are “frequent” itemsets of size 1 (or, “frequent” 1-itemsets).

We set L_1 = {{A}, {B}, {C}, {D}, {E}}

Co-location Mining: Background – Apriori Algorithm

- After the candidate generation (i.e., Join Step and Prune Step), we are given a set of **candidate** itemsets
- We need to **verify** whether these candidate itemsets are frequent or not
- We have to scan the database to obtain the count of each itemset in the candidate set.

Co-location Patterns Mining



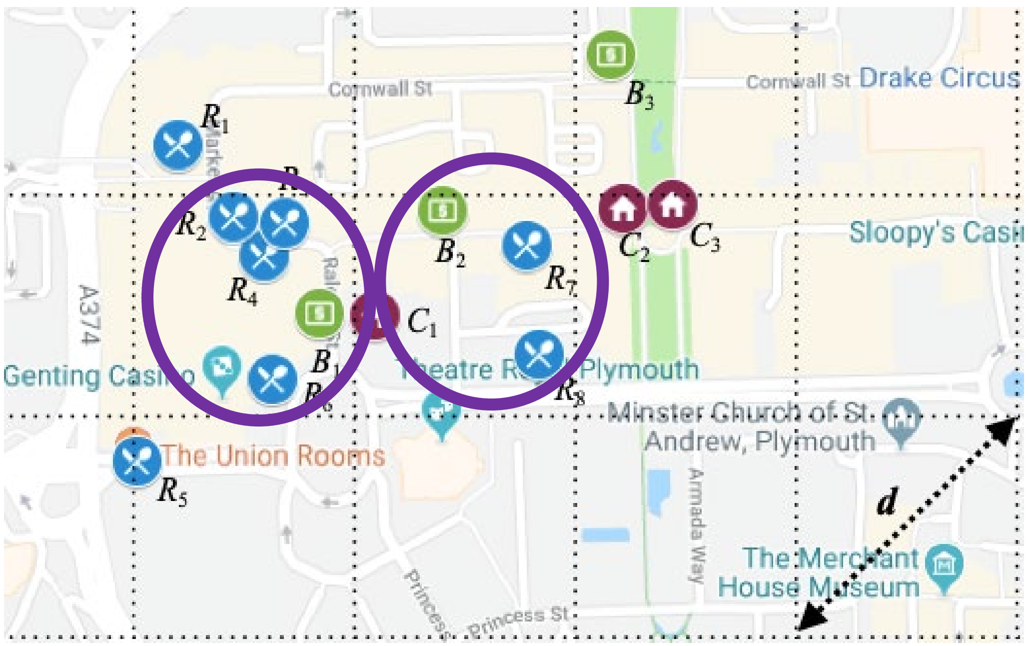
A set of POIs which cover the categories and are located nearby

POI category sets whose instances appear frequently

Support

# of instances?

Co-location Patterns Mining



{Restaurant, Bank}

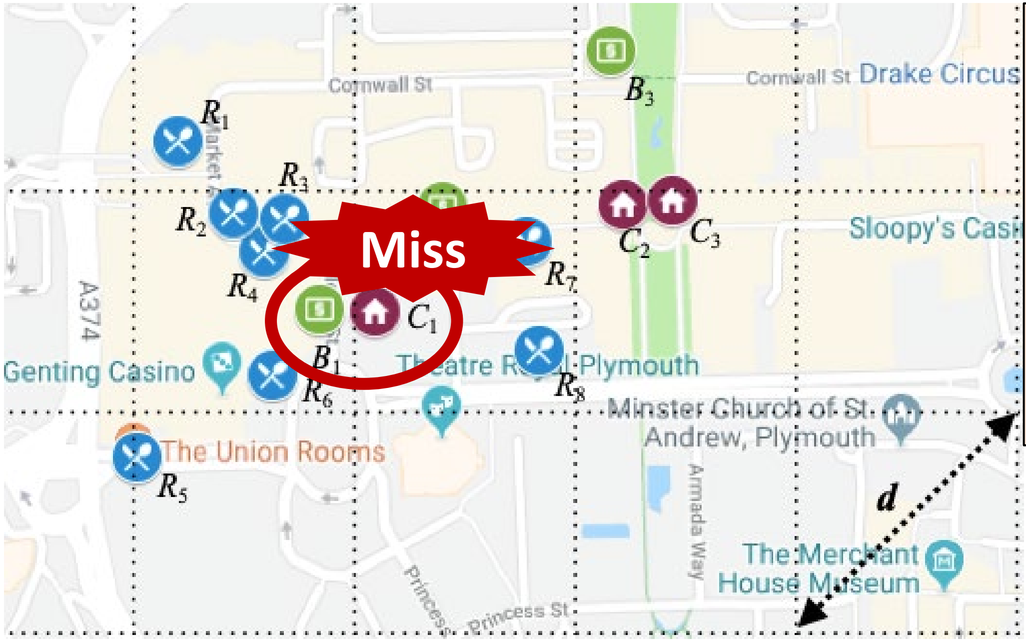
At least 6

{Bank}

3

~~Anti-monotonicity~~

Co-location Patterns Mining –Support Definition (1)

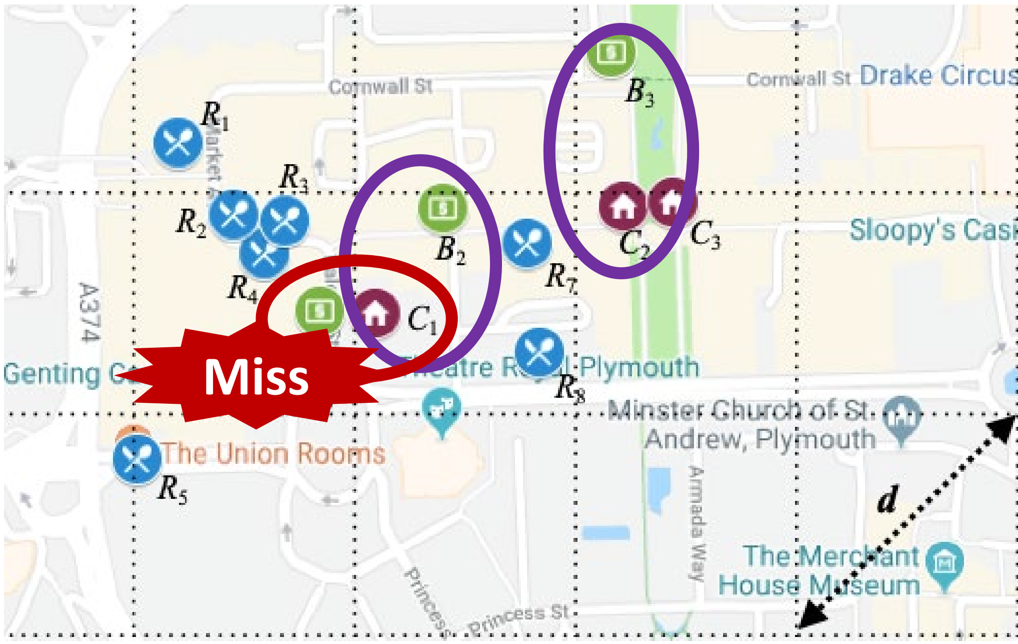


Partitioning-based:

1. Partition the space into grids
2. Treat the POIs within each grid as a "transaction"
3. Define the support **as on the transaction data**

{Church, Bank}

Co-location Patterns Mining –Support Definition (2)

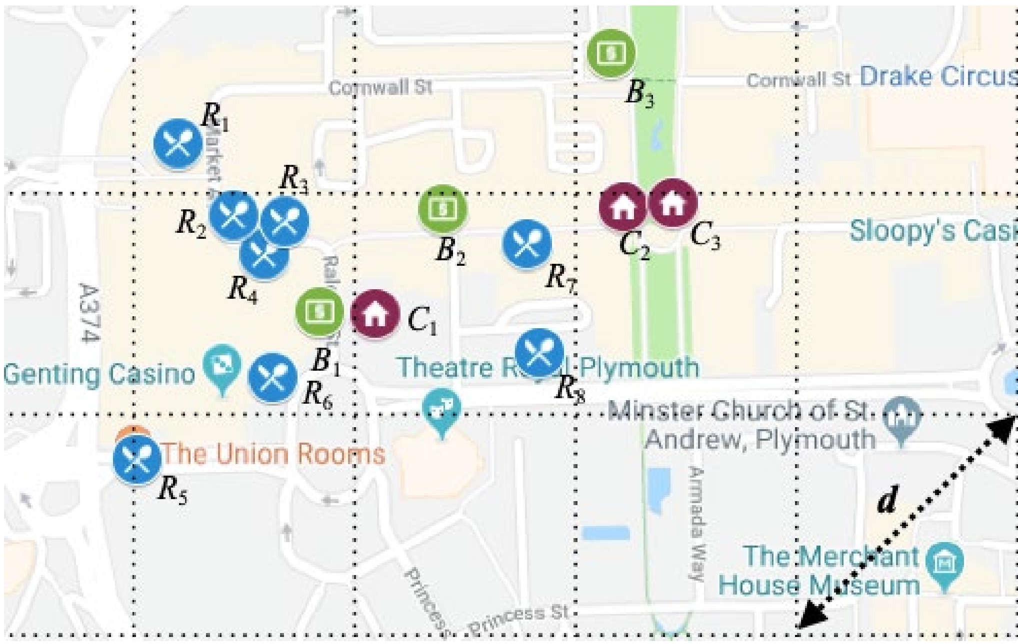


Construction-based:

1. Construct a set of instances **heuristically**
2. Define the support as the **# of constructed instances**

{Bank, Church}

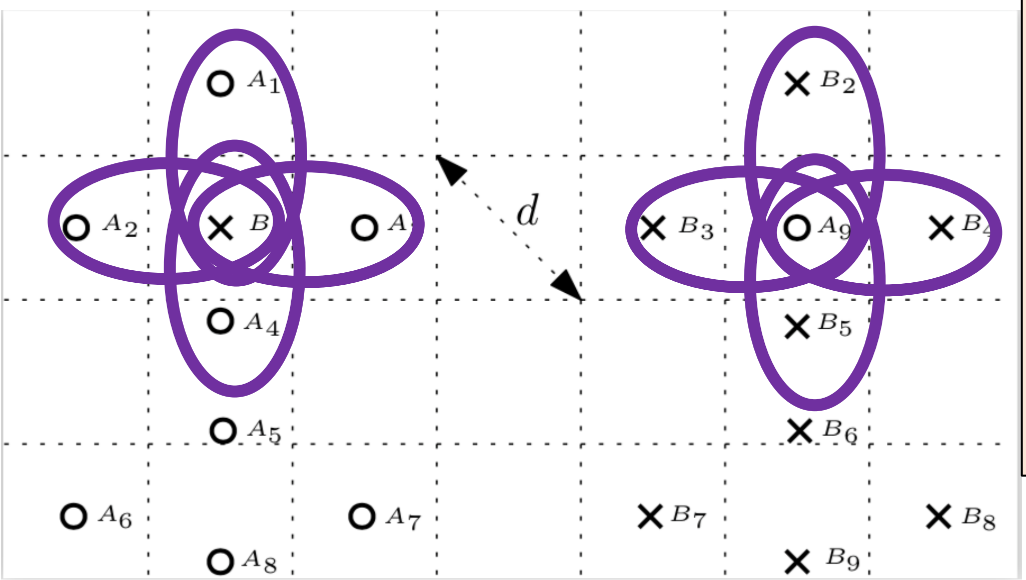
Co-location Patterns Mining –Support Definition (3)



Participation-based:

1. Group the instances sharing a POI (among those with a specified category)
2. Define the support as **the # of groups of instances**

Co-location Patterns Mining –Support Definition (3)



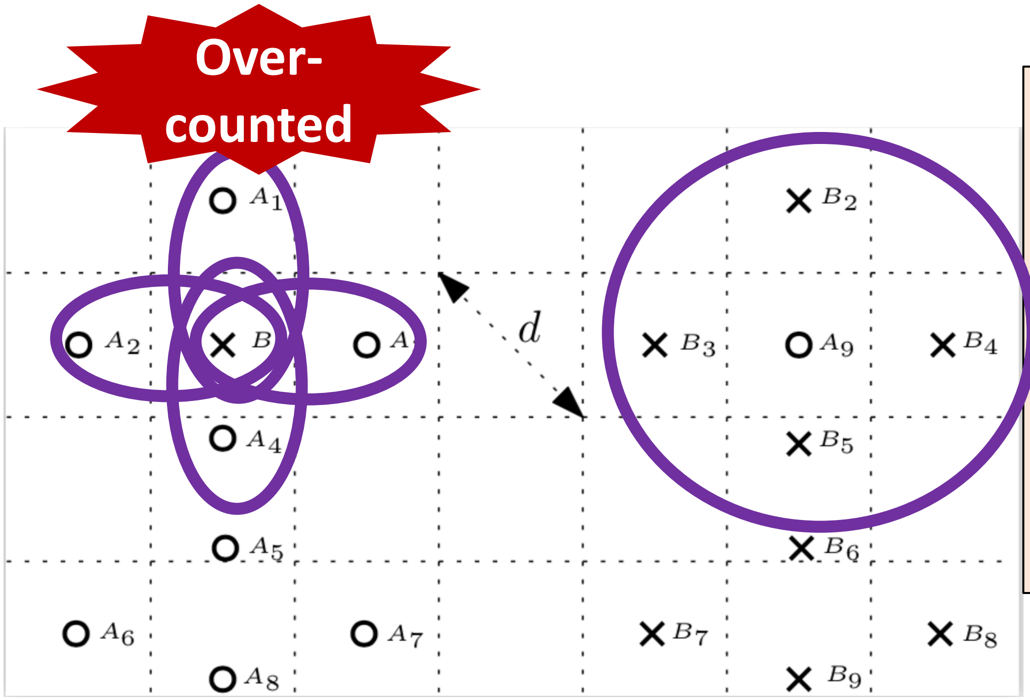
Suppose we specify Category A for grouping

Participation-based:

1. Group the instances sharing a POI (among those with a specified category)
2. Define the support as **the # of groups of instances**

{A, B}

Co-location Patterns Mining –Support Definition (3)



Suppose we specify Category A for grouping

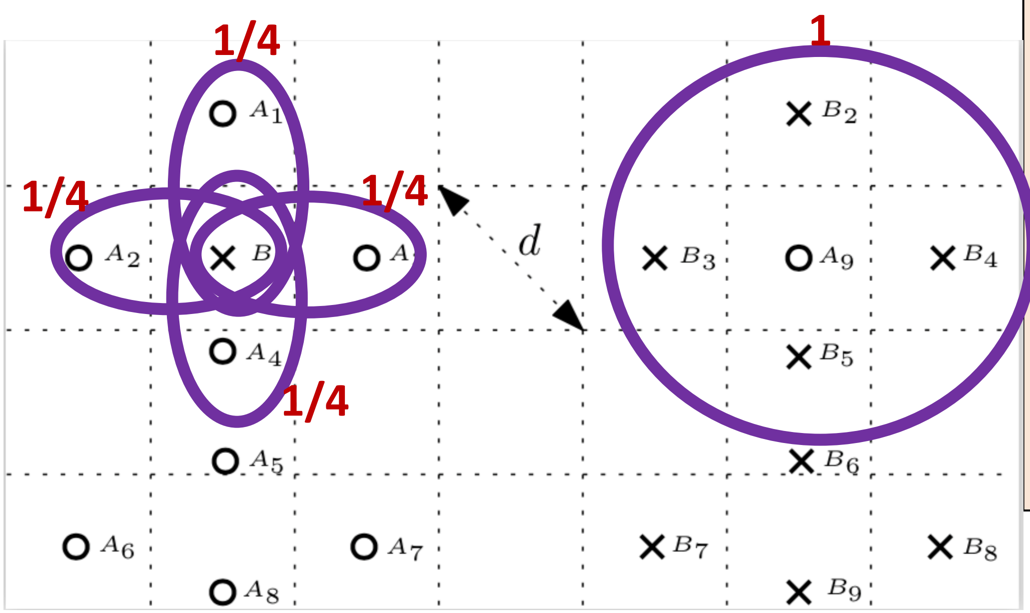
Participation-based:

1. Group the instances sharing a POI (among those with a specified category)
2. Define the support as **the # of groups of instances**

{A, B}

Support = # of groups = 5

Co-location Patterns Mining – Support Definition (4)



Suppose we specify **Category A** for grouping

Fraction-based:

1. Group the instances sharing a POI (among those with a specified category)
2. **Associate each group with a fraction**
3. Define the support as the **sum of the fractions of the groups**

{A, B}

Sum of the fractions of groups = $1/4 \cdot 4 + 1 = 2$

Recap

- Spatial Data Mining
- Spatial Data Clustering (Hotspot)
- Spatial Data Outlier Detection
- Co-location Mining

Next Lecture

Part 2 – 04: Urban Data Learning and Applications
(By Prof Cong Gao)