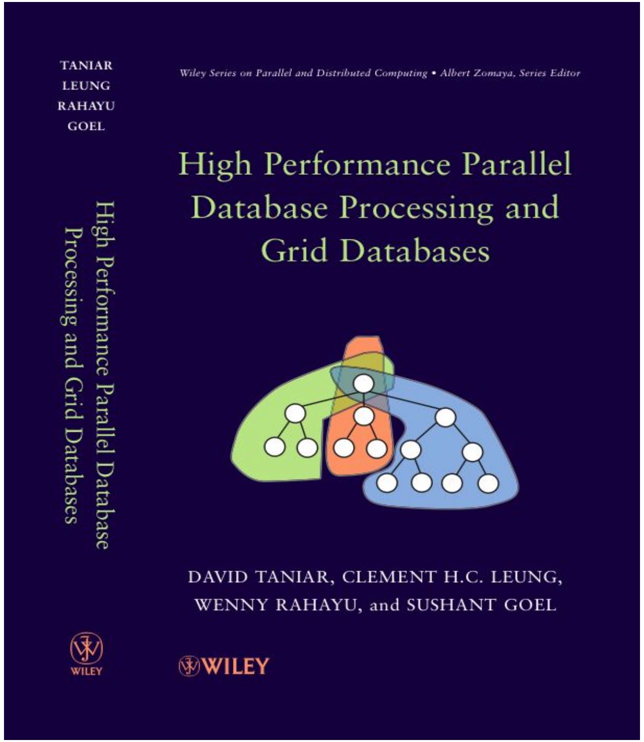
MONASH
INFORMATION
TECHNOLOGY

Machine Learning: Clustering

Prajwol Sangat

This week



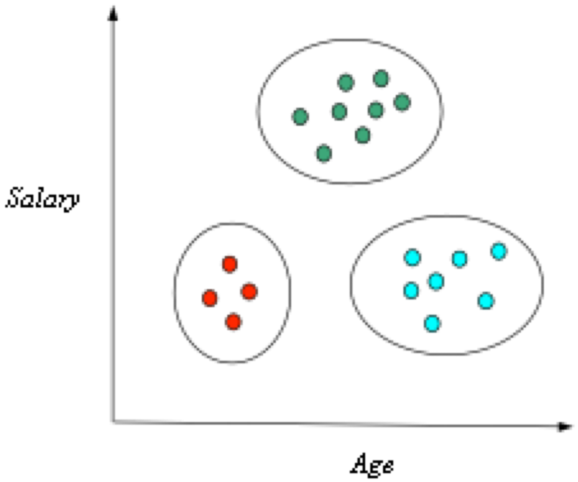
Chapter 17 Parallel Clustering and Classification

Machine Learning Fundamentals - Revision

- Supervised learning vs. unsupervised learning
- Supervised learning: discover patterns in the data that relate to data attributes with a target (class) attribute.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
- Unsupervised learning: The data have no target attribute.
  - Exploring the data to find some intrinsic structures in them.

Clustering: an illustration

- Finds groups (or clusters) of data
- A cluster comprises a number of “similar” objects
- A member is closer to another member within the same group than to a member of a different group
- Groups have no category or labe
- Unsupervised learning



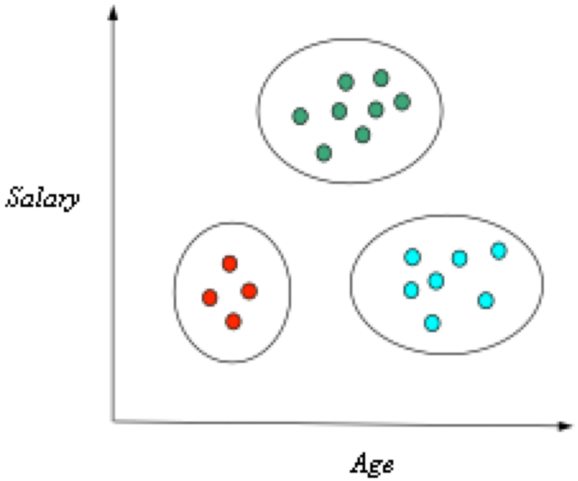
What is clustering for?

- Let’s see some real-life examples
- Example 1: Cluster students based on their examination marks, gender, heights, nationality, etc.

- Example 2: In marketing, segment customers according to their similarities
  - To do targeted marketing.

Clustering: an illustration

- Finds groups (or clusters) of data
- A cluster comprises a number of “similar” objects
- A member is closer to another member within the same group than to a member of a different group
- Groups have no category or lab
- Unsupervised learning



What is clustering for?

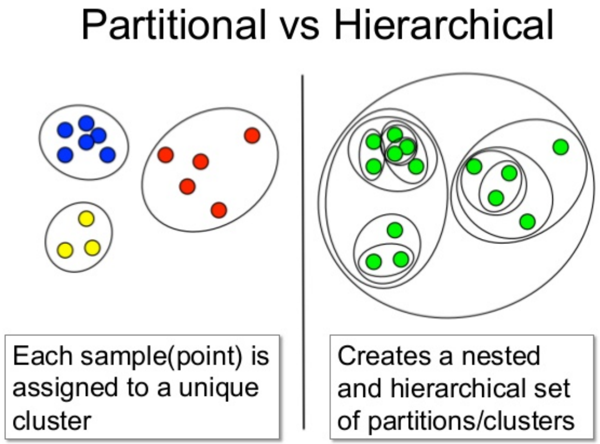
- Clustering is one of the most utilized machine learning techniques.
  - Used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.
  - Most popular applications of clustering are:
    - recommendation engines,
    - market segmentation,
    - social network analysis,
    - image segmentation,
    - anomaly detection

What is clustering for?

- **Similarities Measures**
  - Key factor in clustering is the similarity measure
  - Measure the degree of similarity between two objects
  - Distance measure: the shorter the distance the, the more similar are the two objects (zero distance means identical objects)
  - Euclidean Distance:

$$dist(x_i, x_j) = \sqrt{\sum_{k=1}^{h} \left(x_{ik} - x_{jk}\right)^2}$$

- **Hierarchical** clustering
  - Seeks to build a hierarchy of clust
  - Strategies:
    - *Agglomerative*: Bottom up approach
    - *Divisive*: Top down approach.
- **Partitional** clustering
  - Partitions the data objects based on a clustering criterion.
  - Places the data objects into clusters to maximise intra-cluster similarity.



K-Means clustering (Partitional clustering)

- K-means is a partitional clustering algorithm
- Let a set of data points (or instances) D be

  $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

  where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.
- The k-means algorithm partitions the given data into k clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - k is specified by the user

MONASH University

K-Means clustering

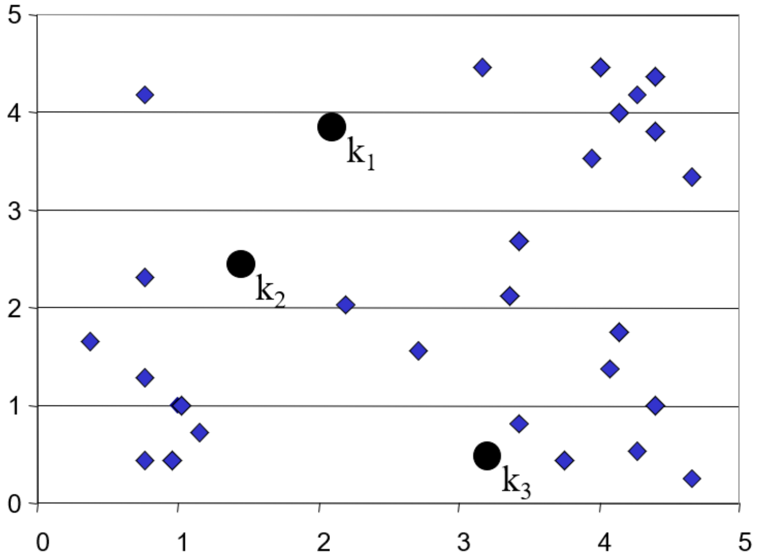
- **Algorithm** k-Means:
  - Specifies *k* number of clusters, and guesses the *k* seed cluster centroid
  - Iteratively looks at each data point and assigns it to the closest centroid
  - Current clusters may receive or loose their members
  - Each cluster must re-calculate the mean (centroid)
  - The process is repeated until the clusters are stable (no change of members)

Algorithm: k-means

Input:
  $D=\{x_1, x_2, \ldots, x_n\}$ //Data objects
  k //Number of desired clusters
Output:
  K //Set of clusters
1. Assign initial values for means $m_1, m_2, \ldots, m_k$
2. Repeat
3. Assign each data object x_i to the cluster which has the closest mean
4. Calculate new mean for each cluster
5. Until convergence criteria is met

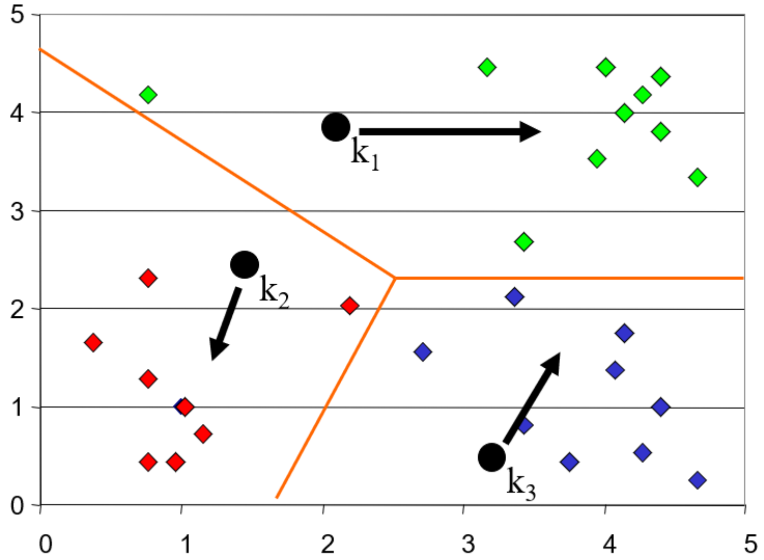
K-Means Clustering: Step 1

- Algorithm: k-means, Distance Metric: Euclidean Distance



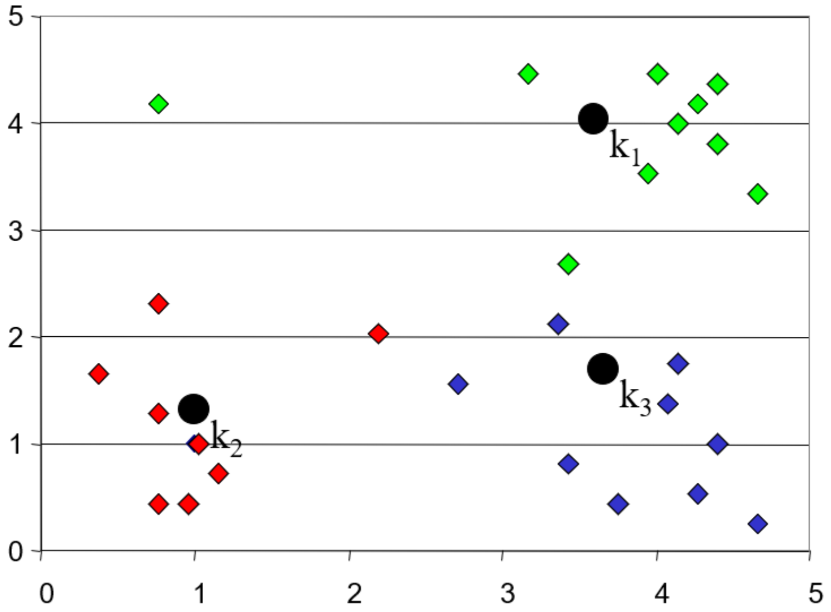
K-Means Clustering: Step 2

- Algorithm: k-means, Distance Metric: Euclidean Distance



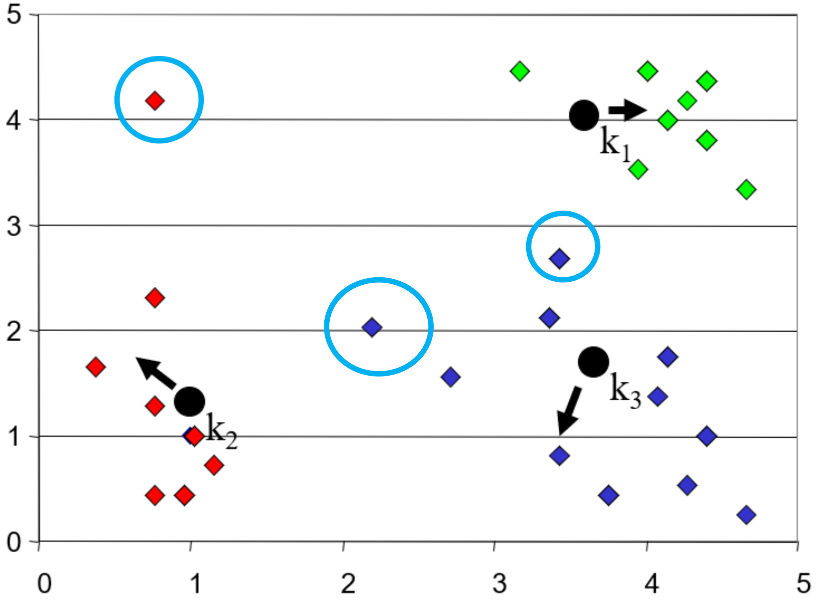
K-Means Clustering: Step 3

- Algorithm: k-means, Distance Metric: Euclidean Distance



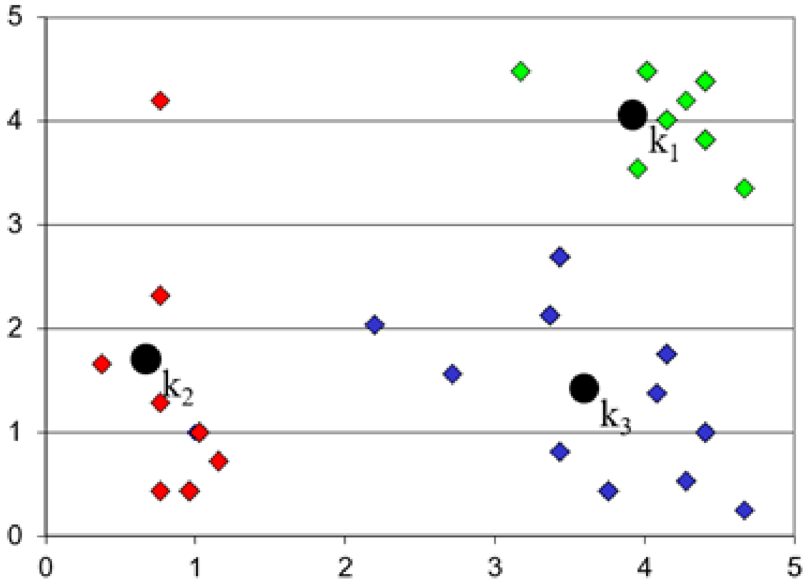
K-Means Clustering: Step 4

- Algorithm: k-means, Distance Metric: Euclidean Distance



K-Means Clustering: Step 5

- Algorithm: k-means, Distance Metric: Euclidean Distance



k-Means: Step-By-Step Example

- Data D = {5, 19, 25, 21, 4, 1, 17, 23, 8, 7, 6, 10, 2, 20, 14, 11, 27, 9, 3, 16}
- Number of clusters: k = 3
- Initial centroids: m_1=6, m_2=7, and m_3=8

- **First Iteration**
  - Clusters:
    - C_1={1, 2, 3, 4, 5, 6}
    - C_2={7}
    - C_3={8, 9, 10, 11, 14, 16, 17, 19, 20, 21, 23, 25, 27}
  - Re-calculated centroids: m_1=3.5, m_2=7, and m_3=16.9

k-Means: Step-By-Step Example

- Data D = {5, 19, 25, 21, 4, 1, 17, 23, 8, 7, 6, 10, 2, 20, 14, 11, 27, 9, 3, 16}
- Number of clusters: $k = 3$
- Initial centroids: m_1=6, m_2=7, and m_3=8

- **First Iteration**
  - Clusters:
    - C_1={1, 2, 3, 4, 5, 6}
    - C_2={7}
    - C_3={8, 9, 10, 11, 14, 16, 17, 19, 20, 21, 23, 25, 27}
  - Re-calculated centroids: m_1=3.5, m_2=7, and m_3=16.9

MONASH University

k-Means: Step-By-Step Example

- Clusters:
  - C_1={1, 2, 3, 4, 5, 6}
  - C_2={7}
  - C_3={8, 9, 10, 11, 14, 16, 17, 19, 20, 21, 23, 25, 27}
- New centroids: m_1=3.5, m_2=7, and m_3=16.9
- **Second Iteration**
  - Clusters:
    - C_1={1, 2, 3, 4, 5}
    - C_2={6, 7, 8, 9, 10, 11}
    - C_3={14, 16, 17, 19, 20, 21, 23, 25, 27}
  - Re-calculated centroids: m_1=3, m_2=8.5, and m_3=20.2

k-Means: Step-By-Step Example

- Clusters:
  - C_1={1, 2, 3, 4, 5}
  - C_2={6, 7, 8, 9, 10, 11}
  - C_3={14, 16, 17, 19, 20, 21, 23, 25, 27}
- New centroids: m_1=3, m_2=8.5, and m_3=20.2
- **Third Iteration**
  - Clusters:
    - C_1={1, 2, 3, 4, 5}
    - C_2={6, 7, 8, 9, 10, 11, 14}
    - C_3={16, 17, 19, 20, 21, 23, 25, 27}
  - Re-calculated centroids: m_1=3, m_2=9.29, and m_3=21

k-Means: Step-By-Step Example

- Clusters:
  - C_1={1, 2, 3, 4, 5}
  - C_2={6, 7, 8, 9, 10, 11, 14}
  - C_3={16, 17, 19, 20, 21, 23, 25, 27}
- New centroids: m_1=3, m_2=9.29, and m_3=21
- **Fourth Iteration**
  - Clusters:
    - C_1={1, 2, 3, 4, 5, 6}
    - C_2={7, 8, 9, 10, 11, 14}
    - C_3={16, 17, 19, 20, 21, 23, 25, 27}
  - Re-calculated centroids: m_1=3.5, m_2=9.83, and m_3=21

k-Means: Step-By-Step Example

- Clusters:
  - C_1={1, 2, 3, 4, 5, 6}
  - C_2={7, 8, 9, 10, 11, 14}
  - C_3={16, 17, 19, 20, 21, 23, 25, 27}
- New centroids: m_1=3.5, m_2=9.83, and m_3=21
- **Fifth Iteration**
  - No data movement from clusters (Process Terminated)

m_1	m_2	m_3	C_1	C_2	C_3
6	7	8	1, 2, 3, 4, 5, 6	7	8, 9, 10, 11, 14, 16, 17, 19, 20, 23, 25, 27
3.5	7	16.9	1, 2, 3, 4, 5	6, 7, 8, 9, 10, 11	14, 16, 17, 19, 20, 21, 23, 25, 27
3	8.5	20.2	1, 2, 3, 4, 5	6, 7, 8, 9, 10, 11, 14	16, 17, 19, 20, 21, 23, 25, 27
3	9.29	21	1, 2, 3, 4, 5, 6	7, 8, 9, 10, 11, 14	16, 17, 19, 20, 21, 23, 25, 27
3.5	9.83	21	1, 2, 3, 4, 5, 6	7, 8, 9, 10, 11, 14	16, 17, 19, 20, 21, 23, 25, 27

K-Means Clustering

- The number of clusters *k* is predefined. The algorithm does not discover the ideal number of clusters. During the process, the number of clusters remains fixed – it does not shrink nor expand.
- The final composition of clusters is very sensitive to the choice of initial centroid values. Different initialisations may result in (

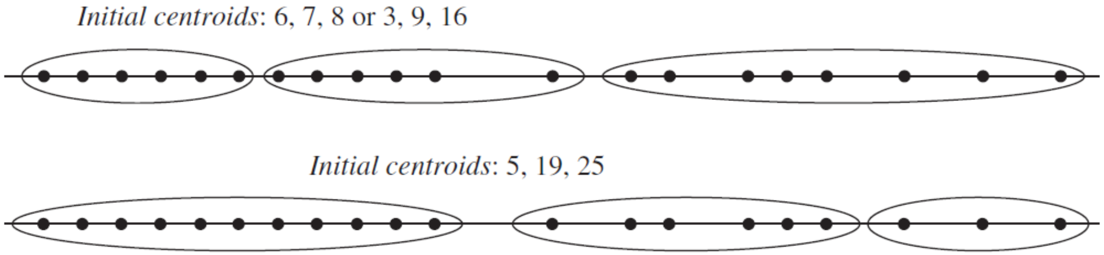


Figure 17.4 Different clustering results for different initial centroids

K-Means Clustering: Pros and Cons

Pros

- Simple and fast for low dimensional data (time complexity of K Means is linear i.e. O(n))
- Scales to large data sets
- Easily adapts to new data points

🕐 **Cons**

- 🕐 It will not identify outliers
- Restricted to data which has the notion of a centre (centroid)

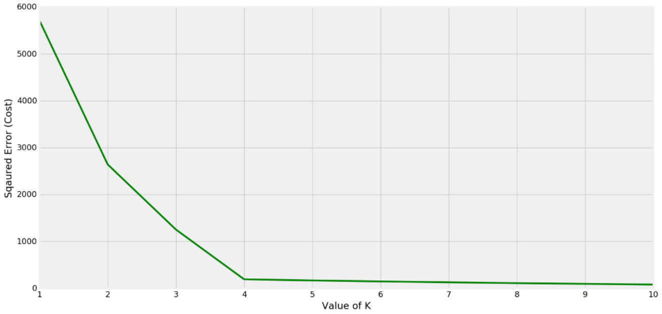
- **Exercise 1**
  - Data $D = \{8, 11, 12, 14, 16, 17, 24, 28\}$
  - Number of clusters: $k = 3$
  - Initial centroids: m_1=11, m_2=12, and m_3=28
  - Use the *k*-means *serial* algorithm to cluster the data in three clusters

Finding Optimal number of the clusters

- As k increases, clusters become smaller.
- The neighbouring clusters become less distinct fr one another.

.

- **How to choose an optimal k?**
  - Elbow Method
    - Sum of squared errors as a function of k (a scree plot)
  - Silhouette analysis
    - Measure of how close each point in one cluster is to points in the neighbouring clusters and thus provides a way to assess number of clus



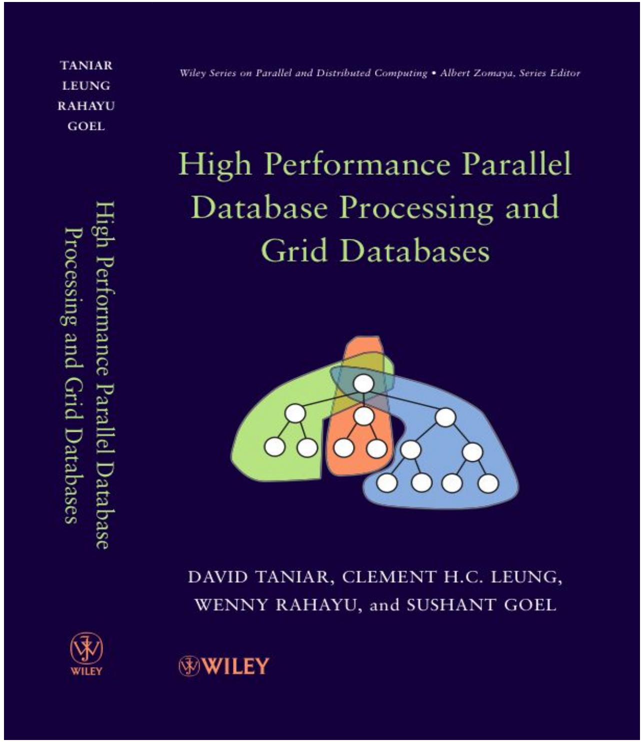
optimal value for k = 4

```
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721
For n_clusters = 4 The average silhouette_score is : 0.6505186632729437
For n_clusters = 5 The average silhouette_score is : 0.56376469026194
For n_clusters = 6 The average silhouette_score is : 0.4504666294372765
```

DEMO

https://instaview365.com/request-a-demo/

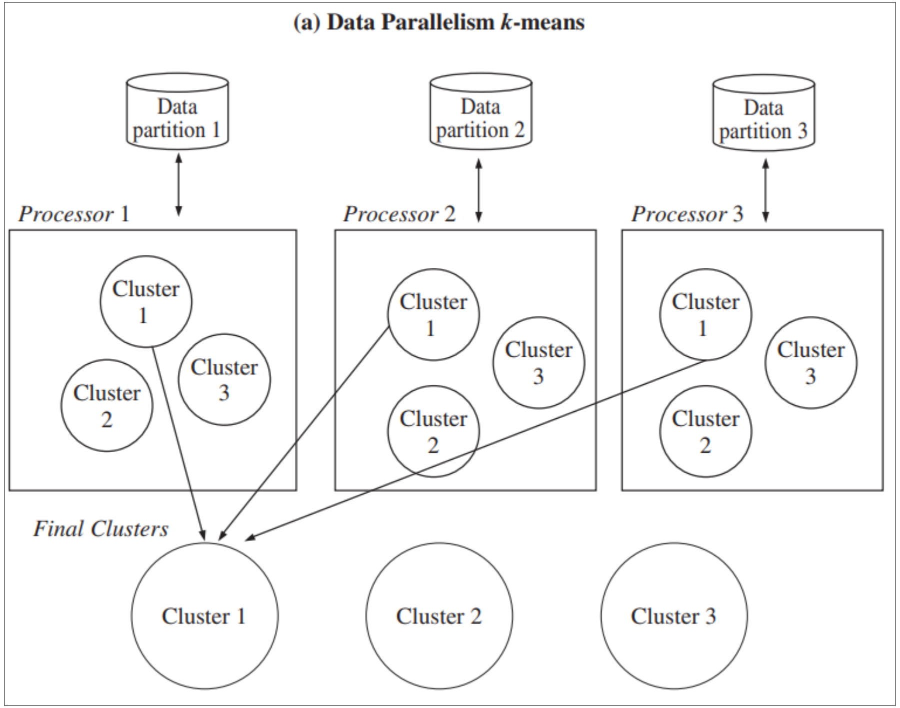
This week



Chapter 17
Parallel Clustering and Classification

Parallel K-means clustering

- ***Data parallelism*** of k-means



Parallel K-means

- ***Data parallelism***

k-means

Initial dataset: 5, 19, 25, 21, 4, 1, 17, 23, 8, 7, 6, 10, 2, 20, 14, 11, 27, 9, 3, 16

Processor 1	**Processor 2**	**Processor 3**
Data partition 1: 5, 21, 17, 7, 2, 11, 3	Data partition 2: 19, 4, 23, 6, 20, 27, 16	Data partition 3: 25, 1, 8, 10, 14, 9

***Iteration* 1**

Cluster 1 Mean=6 Dataset=2, 3, 5 Sum=10; Count=3	Cluster 1 Mean=6 Dataset=4, 6 Sum=10; Count=2	Cluster 1 Mean=6 Dataset=1 Sum=1; Count=1
Cluster 2 Mean=7 Dataset=7 Sum=7; Count=1	Cluster 2 Mean=7 Dataset=NIL Sum=0; Count=0	Cluster 2 Mean=7 Dataset=NIL Sum=0; Count=0
Cluster 3 Mean=8 Dataset=11, 17, 21 Sum=49; Count=3	Cluster 3 Mean=8 Dataset=16, 19, 20, 23, 27 Sum=105; Count=5	Cluster 3 Mean=8 Dataset=8, 9, 10, 14, 25 Sum=66; Count=5

***Iteration* 2**

Cluster 1 Mean=3.5 Dataset=2, 3, 5 Sum=10; Count=3	Cluster 1 Mean=3.5 Dataset=4 Sum=4; Count=1	Cluster 1 Mean=3.5 Dataset=1 Sum=1; Count=1
Cluster 2 Mean=7 Dataset=7, 11 Sum=18; Count=2	Cluster 2 Mean=7 Dataset=6 Sum=6; Count=1	Cluster 2 Mean=7 Dataset=8, 9, 10 Sum=27; Count=3
Cluster 3 Mean=16.92 Dataset=17, 21 Sum=38; Count=2	Cluster 3 Mean=16.92 Dataset=16, 19, 20, 23, 27 Sum=105; Count=5	Cluster 3 Mean=16.92 Dataset=14, 25 Sum=39; Count=2

Parallel K-means

- ***Data parallelism***

k-means

Processor 1: Cluster 1 = 2, 3, 5
Cluster 2 = 7, 11
Cluster 3 = 17, 21

Processor 2: Cluster 1 = 4, 6
Cluster 2 = NIL
Cluster 3 = 16, 19, 20, 23, 27

Processor 3: Cluster 1 = 1
Cluster 2 = 8, 9, 10, 14
Cluster 3 = 25

Cluster 1 = 1, 2, 3, 4, 5, 6
Cluster 2 = 7, 8, 9, 10, 11, 14
Cluster 3 = 16, 17, 19, 20, 21, 23, 25, 27

Initial dataset: 5, 19, 25, 21, 4, 1, 17, 23, 8, 7, 6, 10, 2, 20, 14, 11, 27, 9, 3, 16

Processor 1	**Processor 2**	**Processor 3**
Data partition 1: 5, 21, 17, 7, 2, 11, 3	Data partition 2: 19, 4, 23, 6, 20, 27, 16	Data partition 3: 25, 1, 8, 10, 14, 9

***Iteration* 1**

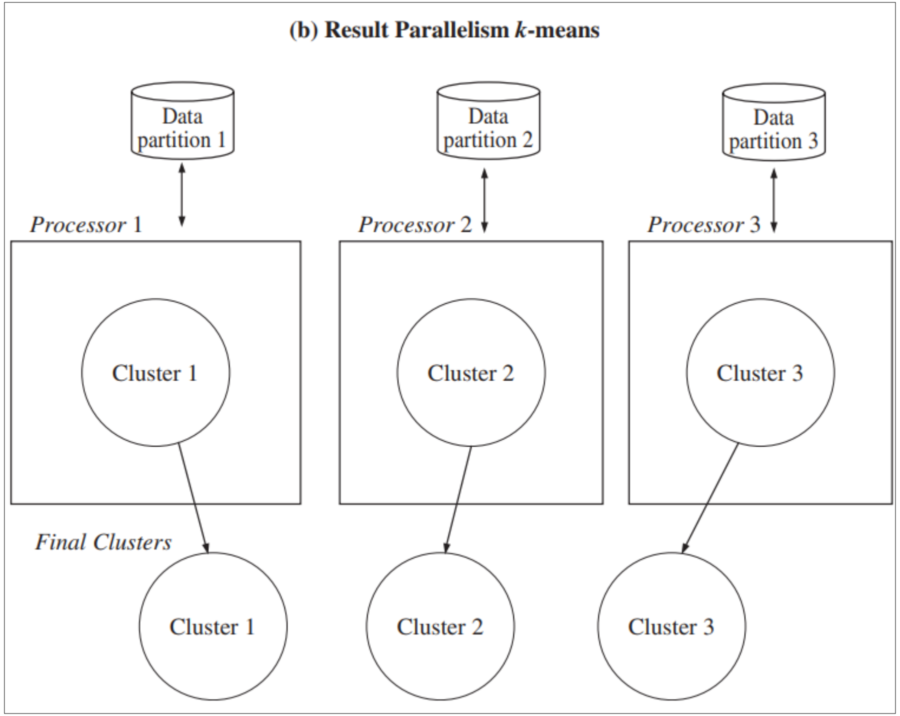
Cluster 1 Mean=6 Dataset=2, 3, 5 Sum=10; Count=3	Cluster 1 Mean=6 Dataset=4, 6 Sum=10; Count=2	Cluster 1 Mean=6 Dataset=1 Sum=1; Count=1
Cluster 2 Mean=7 Dataset=7 Sum=7; Count=1	Cluster 2 Mean=7 Dataset=NIL Sum=0; Count=0	Cluster 2 Mean=7 Dataset=NIL Sum=0; Count=0
Cluster 3 Mean=8 Dataset=11, 17, 21 Sum=49; Count=3	Cluster 3 Mean=8 Dataset=16, 19, 20, 23, 27 Sum=105; Count=5	Cluster 3 Mean=8 Dataset=8, 9, 10, 14, 25 Sum=66; Count=5

***Iteration* 2**

Cluster 1 Mean=3.5 Dataset=2, 3, 5 Sum=10; Count=3	Cluster 1 Mean=3.5 Dataset=4 Sum=4; Count=1	Cluster 1 Mean=3.5 Dataset=1 Sum=1; Count=1
Cluster 2 Mean=7 Dataset=7, 11 Sum=18; Count=2	Cluster 2 Mean=7 Dataset=6 Sum=6; Count=1	Cluster 2 Mean=7 Dataset=8, 9, 10 Sum=27; Count=3
Cluster 3 Mean=16.92 Dataset=17, 21 Sum=38; Count=2	Cluster 3 Mean=16.92 Dataset=16, 19, 20, 23, 27 Sum=105; Count=5	Cluster 3 Mean=16.92 Dataset=14, 25 Sum=39; Count=2

Parallel K-means clustering

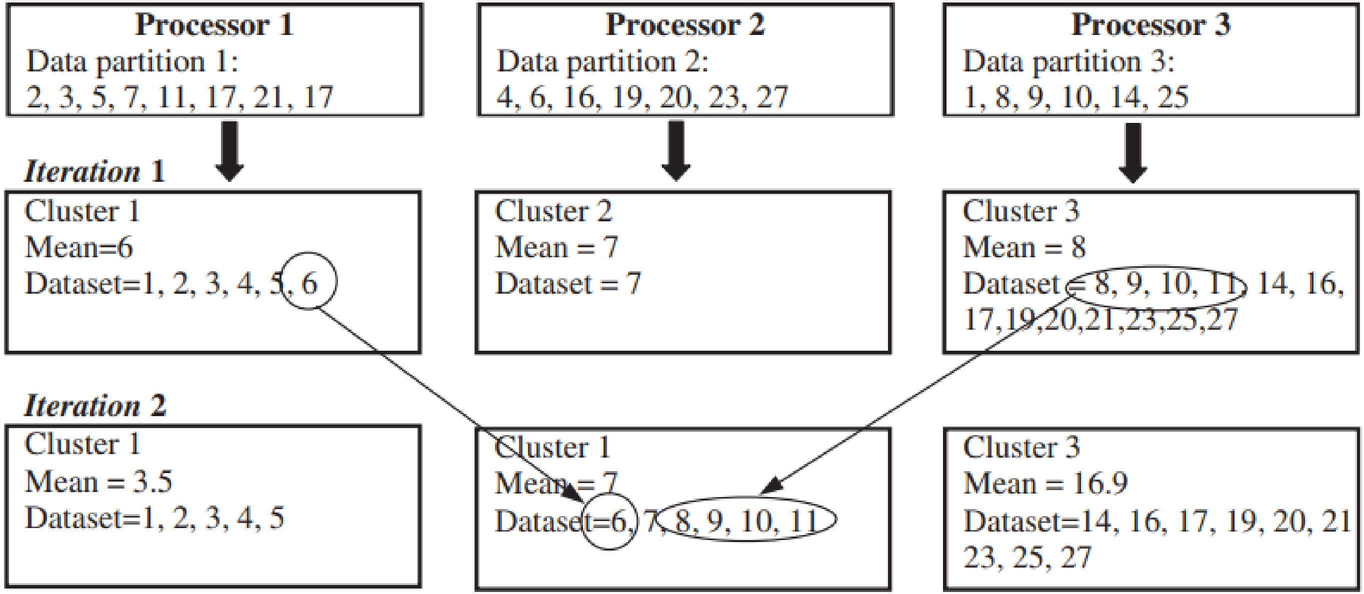
- ***Result Parallelism*** of k-means



Parallel K-means

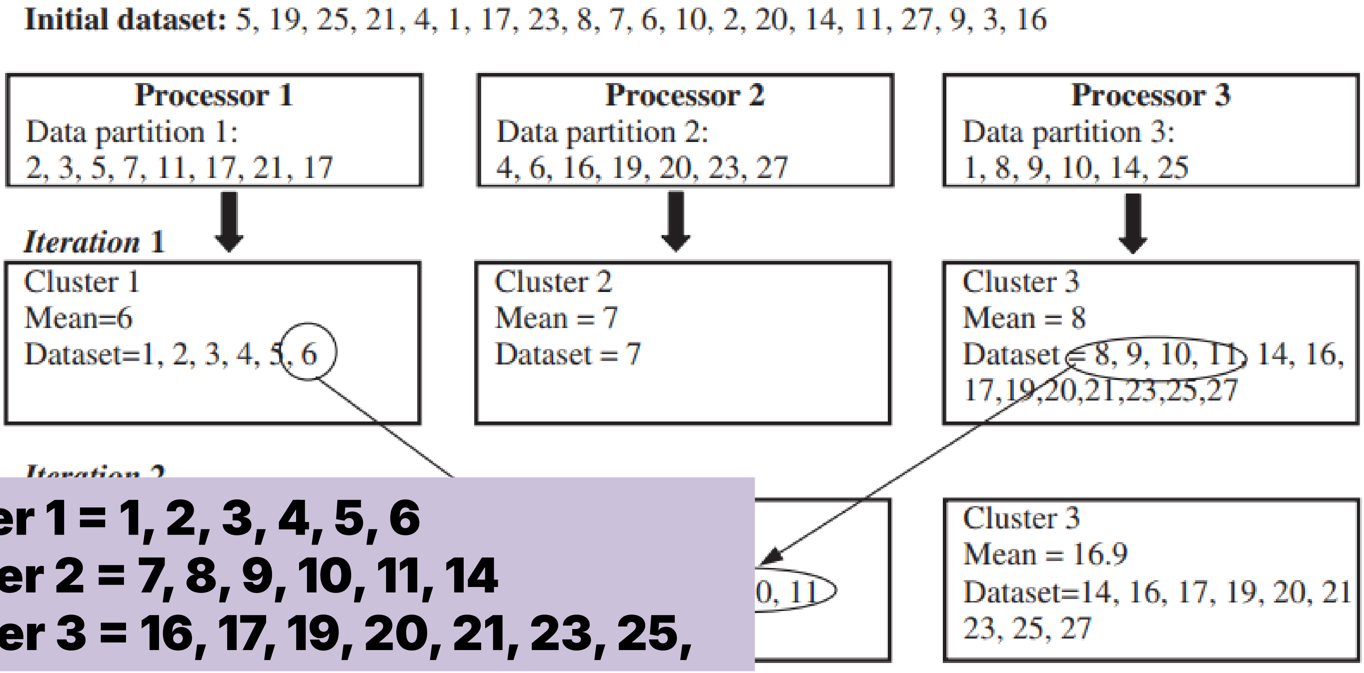
- ***Result parallelism*** k-means

Initial dataset: 5, 19, 25, 21, 4, 1, 17, 23, 8, 7, 6, 10, 2, 20, 14, 11, 27, 9, 3, 16



Parallel K-means

- ***Result parallelism*** k-means



Processor 1 cluster 1 = 1, 2, 3, 4, 5, 6
Processor 2 cluster 2 = 7, 8, 9, 10, 11, 14
Processor 3 cluster 3 = 16, 17, 19, 20, 21, 23, 25, 27

What have we learnt today?

- Partitional (k-means) to attain meaningful groups of data
- Algorithmic examples for clustering of data