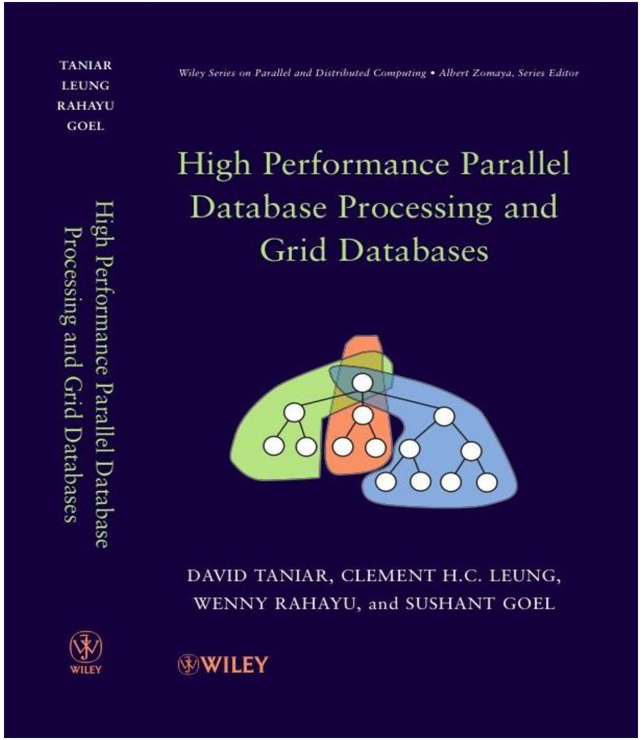
MONASH
INFORMATION
TECHNOLOGY

Machine Learning: Clustering

Prajwol Sangat, Chee-Ming Ting, Lei Yang
Updated by Jay Zhao (Aug/2024)

This week



Chapter 17
Parallel Clustering and Classification

17.1 Clustering and Classification
17.2 Parallel Clustering
17.3 Parallel Classification
17.4 Summary
17.5 Bibliographical Notes
17.6 Exercises

Machine Learning Fundamentals - Revision

- Supervised learning vs. unsupervised learning

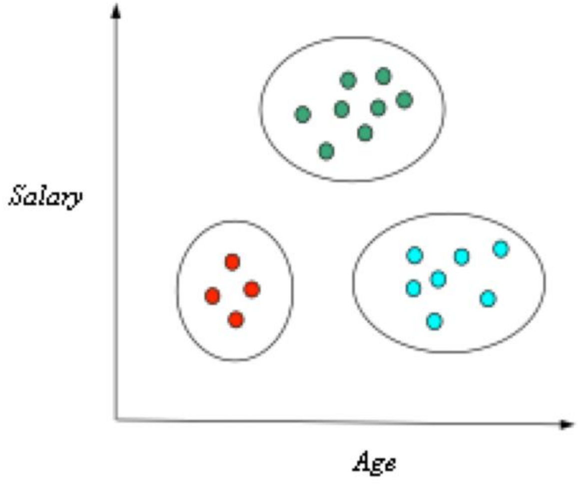
- Supervised learning: discover patterns in the data that relate to data attributes (features) with a target (class) attribute.
  – These patterns are then utilized to predict the values of the target (class) attribute in future data instances.

- 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 (data points are similar within a cluster, less similar between clusters)
- Groups have no category or label
- Unsupervised learning

Definition:
Membership of a data point
- Indicates which cluster a point belong to



sub	salary	age
1		
2		
3		
4		
⋮		
50		

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

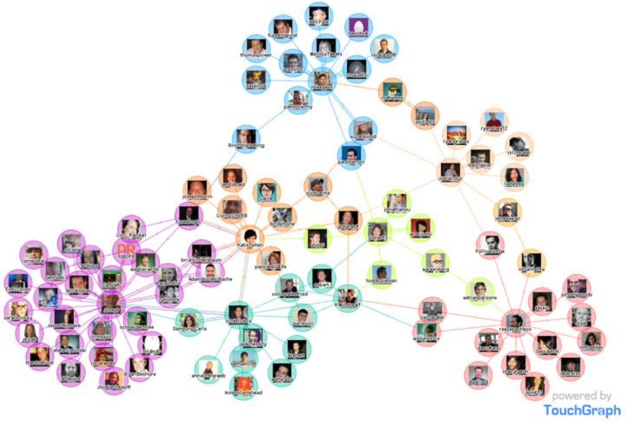


Image from: http://www.touchgraph.com/news

Some Applications in Digital Health

Partitioning of Heart sound signals

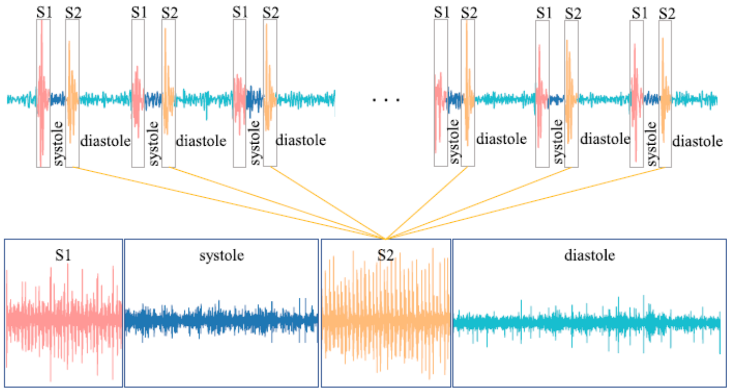
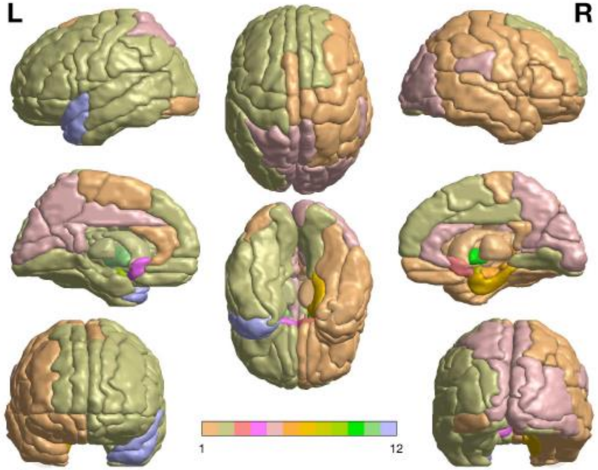


Fig. 2. Dynamic clustering of heart sound into four fundamental components.

Noman, Fuad, Sh-Hussain Salleh, Chee-Ming Ting. "A markov-switching model approach to heart sound segmentation and classification." *IEEE Journal of Biomedical and Health Informatics* 24, no. 3 (2019).

Partitioning of brain regions into clusters (communities)



Ting, Chee-Ming, et al. "Detecting Dynamic Community Structure in Functional Brain Networks Across Individuals: A Multilayer Approach." *IEEE Trans Medical Imaging* (2020).

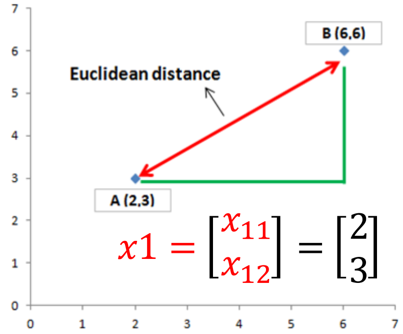
What is clustering for?

$$x2 = \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \end{bmatrix}$$

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 more similar are the two objects (zero distance means identical objects)
- Euclidean Distance:

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



$$x1 = \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$\textit{Euclidean distance}\ (a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

d(x1,x2)

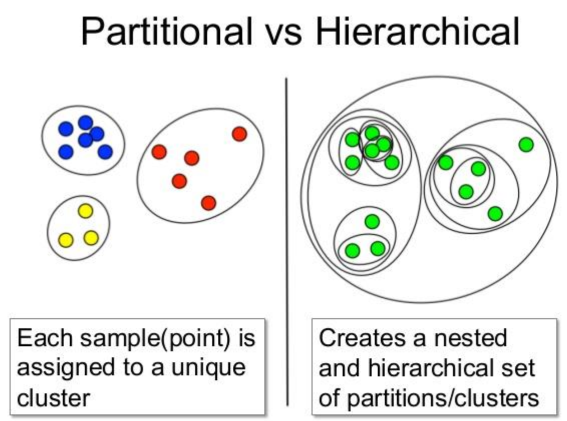
$$= \sqrt{(x_{11} - x_{12})^2 + (x_{21} - x_{22})^2}$$

$$= \sqrt{(2 - 6)^2 + (3 - 6)^2}$$

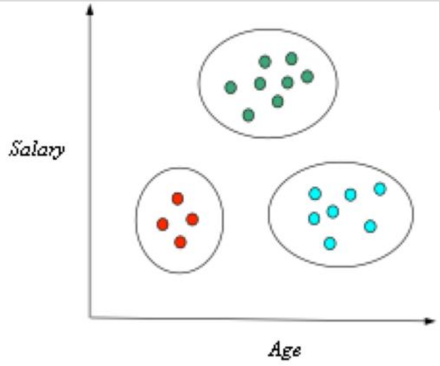
h = number of features (or attributes)

Clustering Techniques

- **Goal of clustering:**
  - maximize intra-cluster similarity & minimize inter-cluster similarity
- **Hierarchical** clustering (nested clustering)
  - Seeks to build a hierarchy of clusters (clusters within clusters)
  - Strategies:
    - *Agglomerative*: Bottom-up approach
    - *Divisive*: Top-down approach.
- **Partitional** clustering (non-overlapping clustering)
  - Partitions the data objects based on a clustering criterion.
  - Places the data objects into clusters to maximise intra-cluster similarity.
  - So that data in a cluster are more similar to each other than to data in different clusters



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**.
  - Centroid can be mean/average of member data points in each cluster
  - k is specified by the user

sub	salary	age
x1		
x2		
x3		
x4		
⋮		
x50		

K-Means clustering

- **Algorithm** k-Means:
  - (**Initialization**) Specifies *k* number of clusters, and guesses the *k* seed cluster centroid
  - (**Assignment Step**) Assign each data point to the cluster with the closest centroid
    - Current clusters may receive or loose their members
  - (**Update Step**) Each cluster must re-calculate the mean (centroid) based on the newly assigned members
  - 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

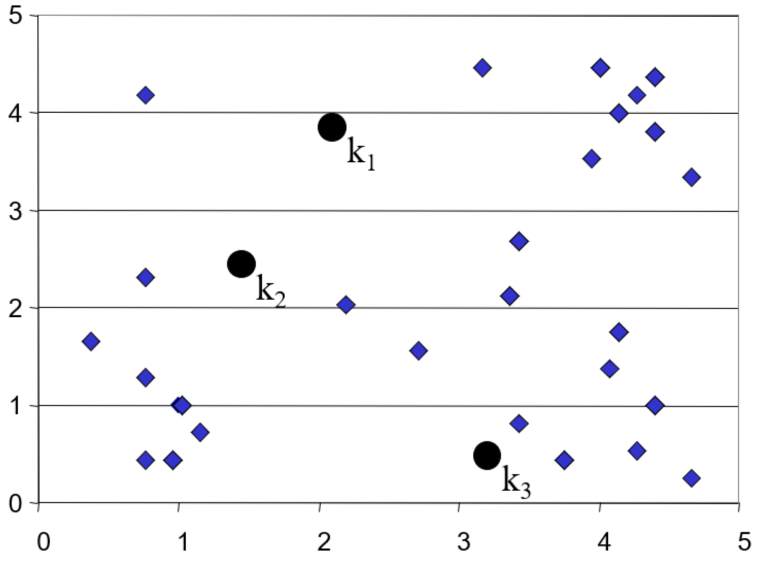
Output:

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

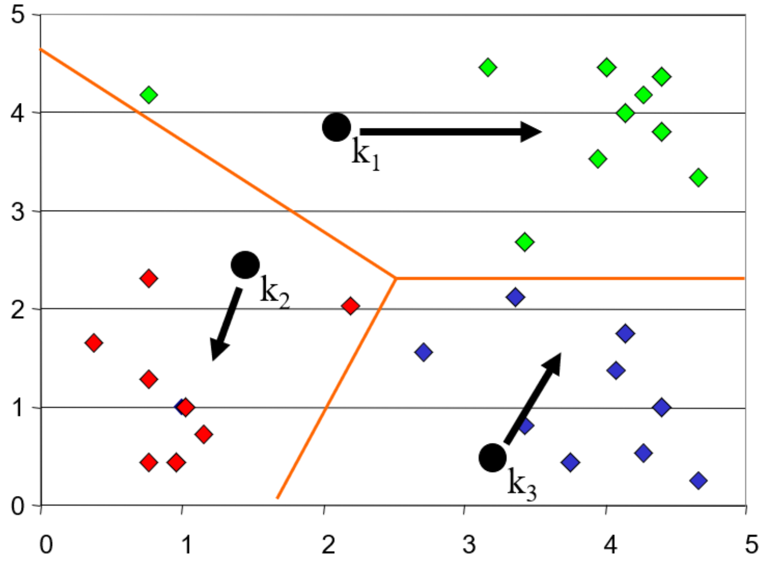
- Algorithm: k-means, Distance Metric: Euclidean Distance



- (**Initialization**)
  Specifies *k* number of clusters, and guesses the *k* seed cluster centroid

K-Means Clustering: Step 2

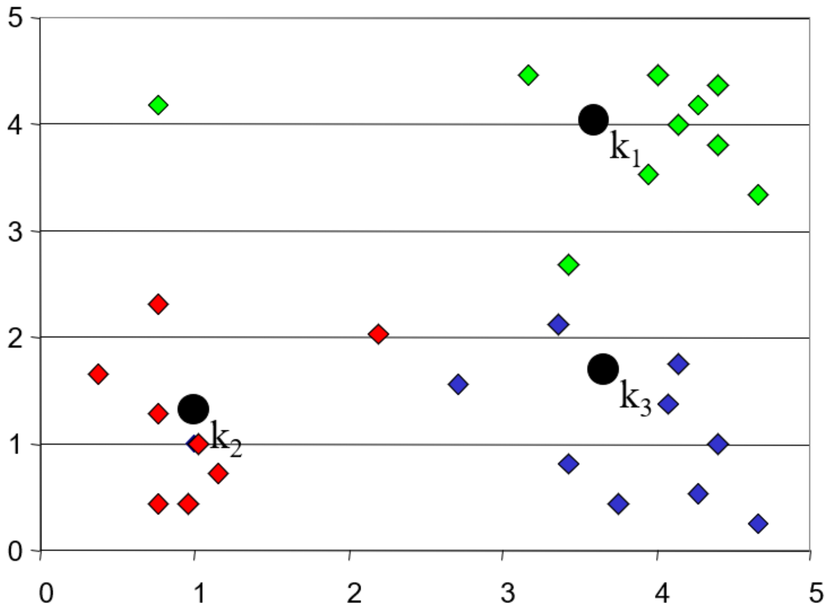
- Algorithm: k-means, Distance Metric: Euclidean Distance



(**Assignment Step**) Iteratively looks at each data point and assigns it to the closest centroid (based on the smallest Euclidean distance)

K-Means Clustering: Step 3

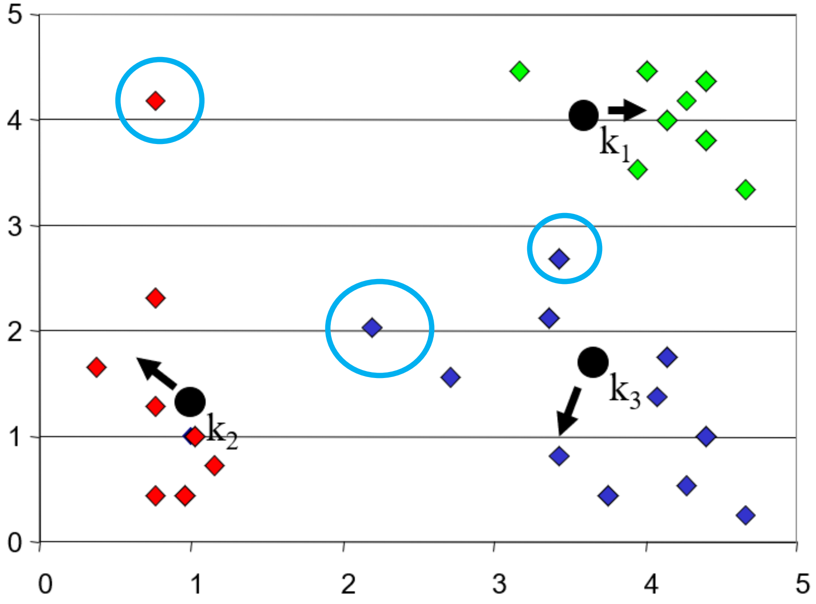
- Algorithm: k-means, Distance Metric: Euclidean Distance



(**Update Step**) Re-calculate the mean (centroid) for each cluster based on the membership of the cluster

K-Means Clustering: Step 4

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

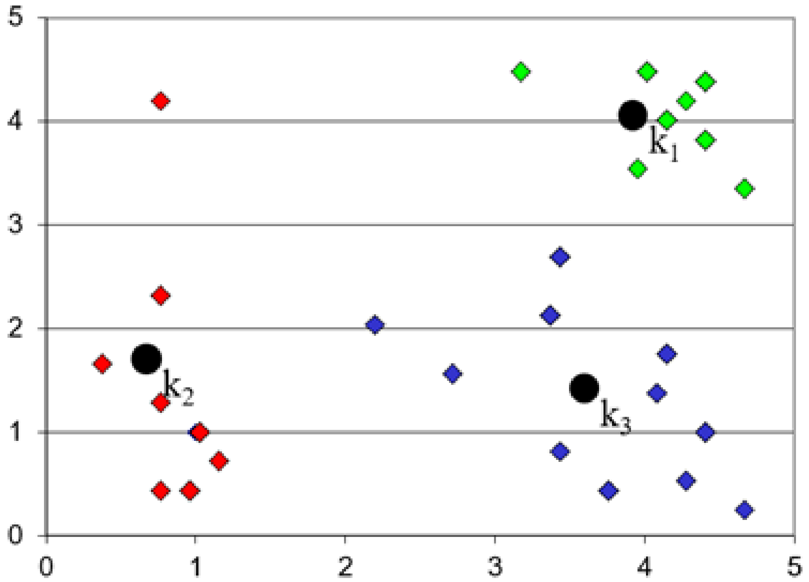


Iteratively looks at each data point and assigns it to the closest centroid,

Current clusters may receive or loose their members

K-Means Clustering: Step 5

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



Re-calculate the mean (centroid) for each cluster based on the membership of the cluster

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**
  - (**Assignment Step)** 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}
  - (**Update Step**) Re-calculated centroids: m_1=3.5, m_2=7, and m_3=16.9

First Iteration: Calculating euclidean distance, determining the cluster membership and calculating new centroid.

D	5	19	25	21	4	1	17	23	8	7	6	10	2	20	14	11	27	9	3	16
d(m1, Di)	1	13	19	15	2	5	11	17	2	1	0	4	4	14	8	5	21	3	3	10
d(m2, Di)	2	12	18	14	3	6	10	16	1	0	1	3	5	13	7	4	20	2	4	9
d(m3, Di)	3	11	17	13	4	7	9	15	0	1	2	2	6	12	6	3	19	1	5	8

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

Second Iteration: Calculating euclidean distance, determining the cluster membership and calculating new centroid.

D	5	19	25	21	4	1	17	23	8	7	6	10	2	20	14	11	27	9	3	16
d(m1, Di)	1.5	15.5	21.5	17.5	0.5	2.5	13.5	19.5	4.5	3.5	2.5	6.5	1.5	16.5	10.5	7.5	23.5	5.5	0.5	12.5
d(m2, Di)	2	12	18	14	3	6	10	16	1	0	1	3	5	13	7	4	20	2	4	9
d(m3, Di)	11.9	2.1	8.1	4.1	12.9	15.9	0.1	6.1	8.9	9.9	10.9	6.9	14.9	3.1	2.9	5.9	10.1	7.9	13.9	0.9

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

Third Iteration: Calculating euclidean distance, determining the cluster membership and calculating new centroid.

D	5	19	25	21	4	1	17	23	8	7	6	10	2	20	14	11	27	9	3	16
d(m1, Di)	2	16	22	18	1	2	14	20	5	4	3	7	1	17	11	8	24	6	0	13
d(m2, Di)	3.5	10.5	16.5	12.5	4.5	7.5	8.5	14.5	0.5	1.5	2.5	1.5	6.5	11.5	5.5	2.5	18.5	0.5	5.5	7.5
d(m3, Di)	15.2	1.2	4.8	0.8	16.2	19.2	3.2	2.8	12.2	13.2	14.2	10.2	18.2	0.2	6.2	9.2	6.8	11.2	17.2	4.2

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

Fourth Iteration: Calculating euclidean distance, determining the cluster membership and calculating new centroid.

D	5	19	25	21	4	1	17	23	8	7	6	10	2	20	14	11	27	9	3	16
d(m1, Di)	2	16	22	18	1	2	14	20	5	4	3	7	1	17	11	8	24	6	0	13
d(m2, Di)	4.3	9.7	15.7	11.7	5.3	8.3	7.7	13.7	1.3	2.3	3.3	0.7	7.3	10.7	4.7	1.7	17.7	0.3	6.3	6.7
d(m3, Di)	16.0	2.0	4.0	0.0	17.0	20.0	4.0	2.0	13.0	14.0	15.0	11.0	19.0	1.0	7.0	10.0	6.0	12.0	18.0	5.0

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

Evaluating K-Means Clusters

- One common measure is sum of squared error (SSE)

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

SSE measures how close the assigned data points to centroids

- small value → maximized intra-cluster similarity

- x is a data point in cluster C_i
- m_i is the centroid of cluster C_i

Example: How to calculate SSE?

Fifth Iteration: Calculating euclidean distance, determining the cluster membership and calculating new centroid.

D	5	19	25	21	4	1	17	23	8	7	6	10	2	20	14	11	27	9	3	16
d(m1, Di)	1.5	15.5	21.5	17.5	0.5	2.5	13.5	19.5	4.5	3.5	2.5	6.5	1.5	16.5	10.5	7.5	23.5	5.5	0.5	12.5
d(m2, Di)	4.8	9.2	15.2	11.2	5.8	8.8	7.2	13.2	1.8	2.8	3.8	0.2	7.8	10.2	4.2	1.2	17.2	0.8	6.8	6.2
d(m3, Di)	16.0	2.0	4.0	0.0	17.0	20.0	4.0	2.0	13.0	14.0	15.0	11.0	19.0	1.0	7.0	10.0	6.0	12.0	18.0	5.0

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}
centroids: m_1=3.5, m_2=9.83, and m_3=21

$$\sum_{x \in C_1} d^2(m_1, x)$$

$$+$$

$$\sum_{x \in C_2} d^2(m_2, x)$$

$$+$$

$$\sum_{x \in C_3} d^2(m_3, x)$$

K-means clustering

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

MONASH University

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 different final clusters composition.

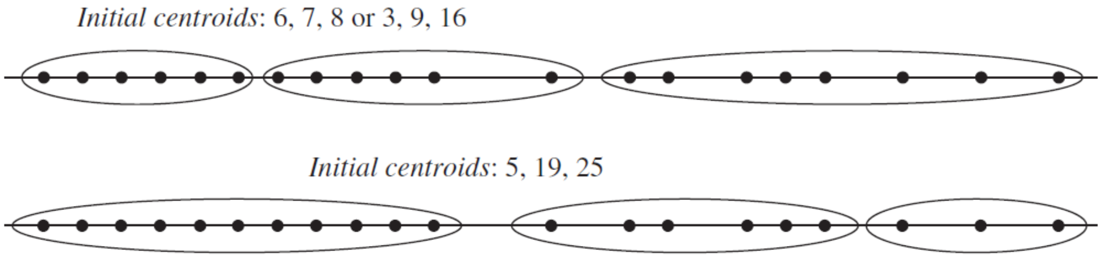


Figure 17.4 Different clustering results for different initial centroids

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

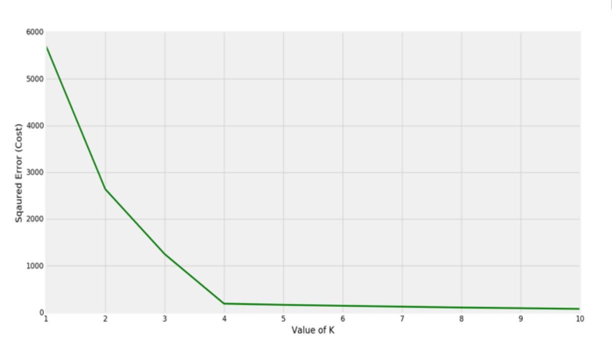
- It will not identify outliers
- Restricted to data which has the notion of a centre (centroid)

Finding Optimal number of the clusters

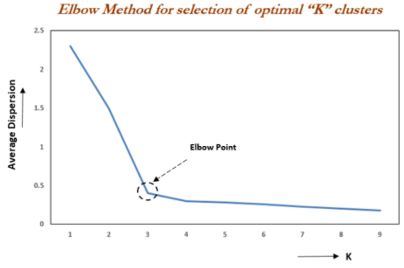
- As k increases, clusters become smaller.
- The neighbouring clusters become less distinct from one another.

.

- **How to choose an optimal k?**
  - Elbow Method
    - Plot sum of squared errors as a function of k (a scree plot)
    - Select the value of k at the “elbow” ie the point after which the SSE start decreasing in a linear fashion.
  - Silhouette analysis
    - Measure of how close each point in one cluster is compared to points in the neighbouring clusters and provides a way to assess number of clusters.
    - If most points have a high silhouette value, then the clustering configuration is appropriate.
    - If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.



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
```

Silhouette Score

Silhouette Score [-1 1] :

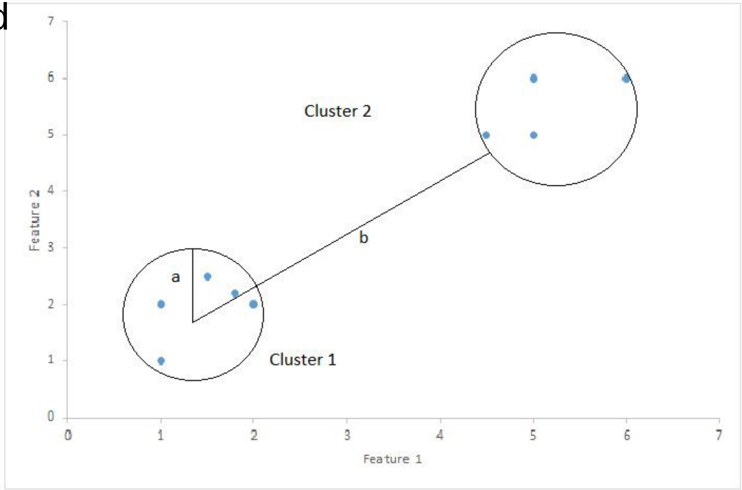
calculates the goodness of a clustering technique

1 - Clusters are well apart from each other and clearly distinguished

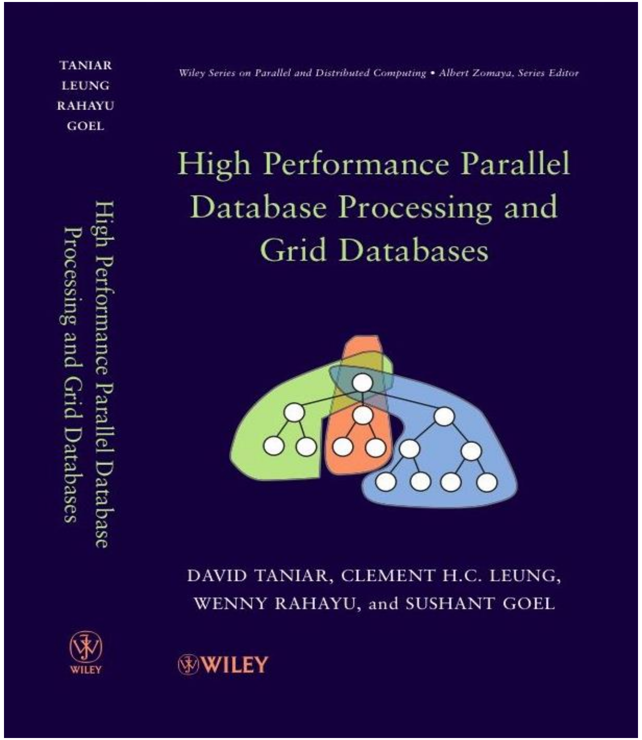
0 - Clusters are not clearly distinguished, the distance between the clusters is not significant (overlapping cluster)

-1 – Clusters assigned wrongly

$$\text{Silhouette Score} = (b-a)/\max(a,b)$$



This week



Chapter 17
Parallel Clustering and Classification

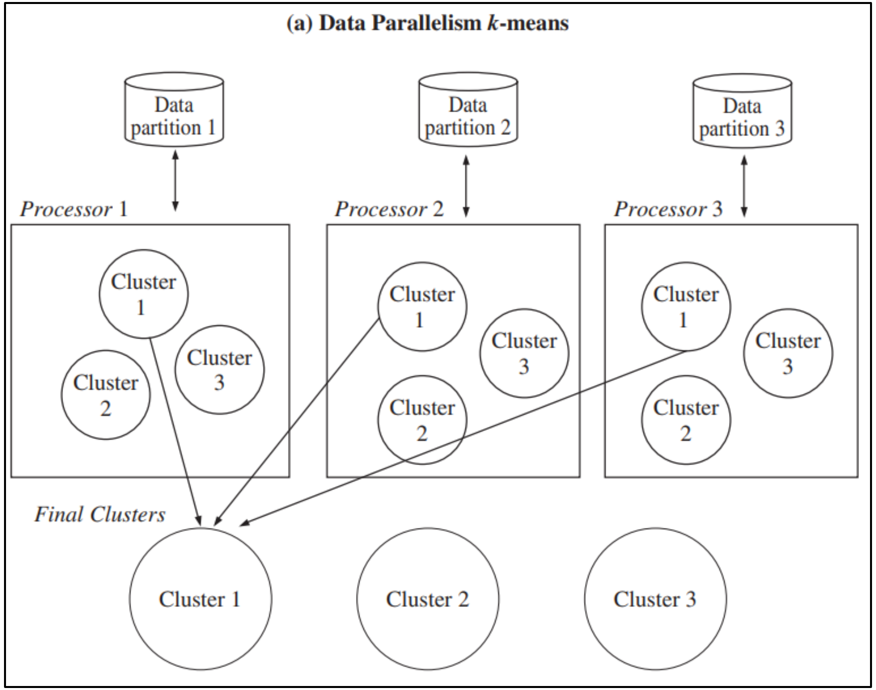
17.1 Clustering and Classification
17.2 Parallel Clustering
17.3 Parallel Classification
17.4 Summary
17.5 Bibliographical Notes
17.6 Exercises

MONASH University

Parallel K-means clustering

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

- ❑ Create parallelism from the beginning because of partitioning of the dataset.
- ❑ Data is partitioned into multiple partition
- ❑ Each processor will work independently to create three clusters (e.g. using k-means algorithm)
- ❑ The final clusters from each processor are respectively united



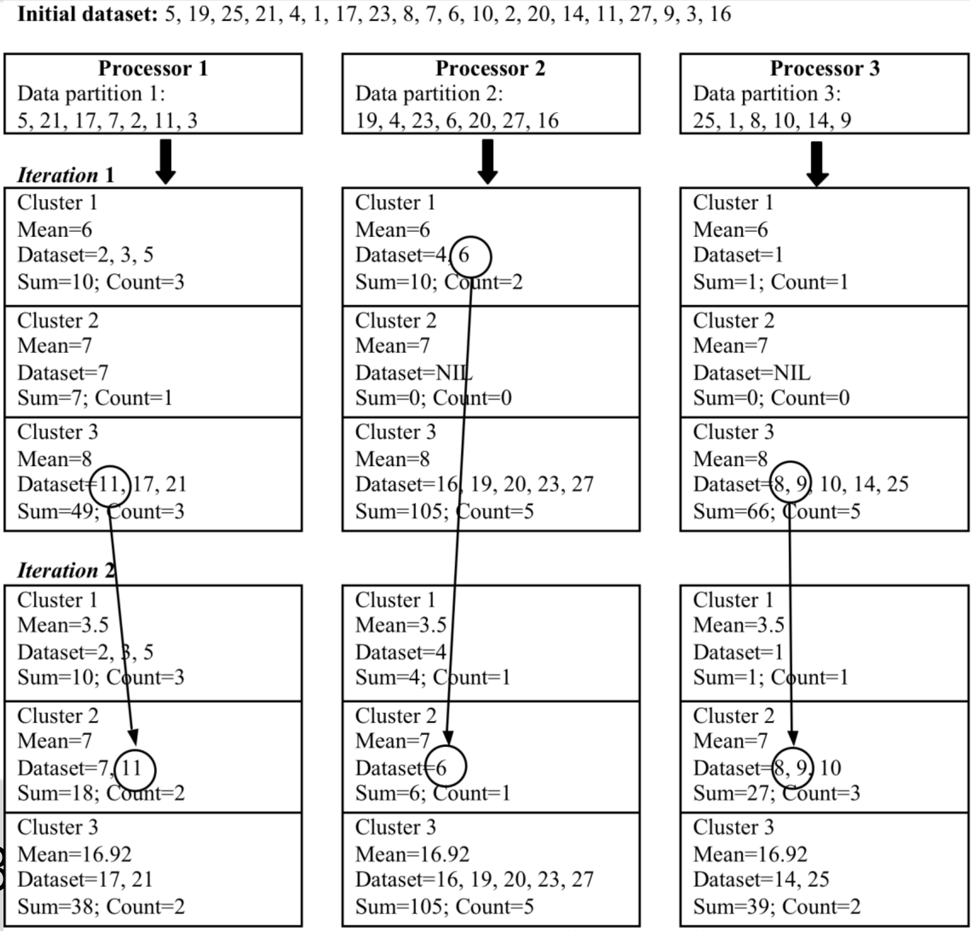
Parallel K-means

- ***Data parallelism***

- Example: Data partitioning using round-robin
- Initial centroids: 6, 7, 8
- Each processor will run k_-Means locally
- At the end of each iteration, info about sum & count of data points in each local cluster is shared between processors to calculate new centroid/mean
- Data does not move among processors (it stays where it was allocated initially)
- Data move across clusters within same processor

Example: Updated centroid of cluster 1 at iter 1:

$$\frac{Total\ sums}{Total\ counts} = \frac{10+10+1}{3+2+1} = \frac{21}{6} = 3.5$$



Parallel K-means

- ***Data parallelism***

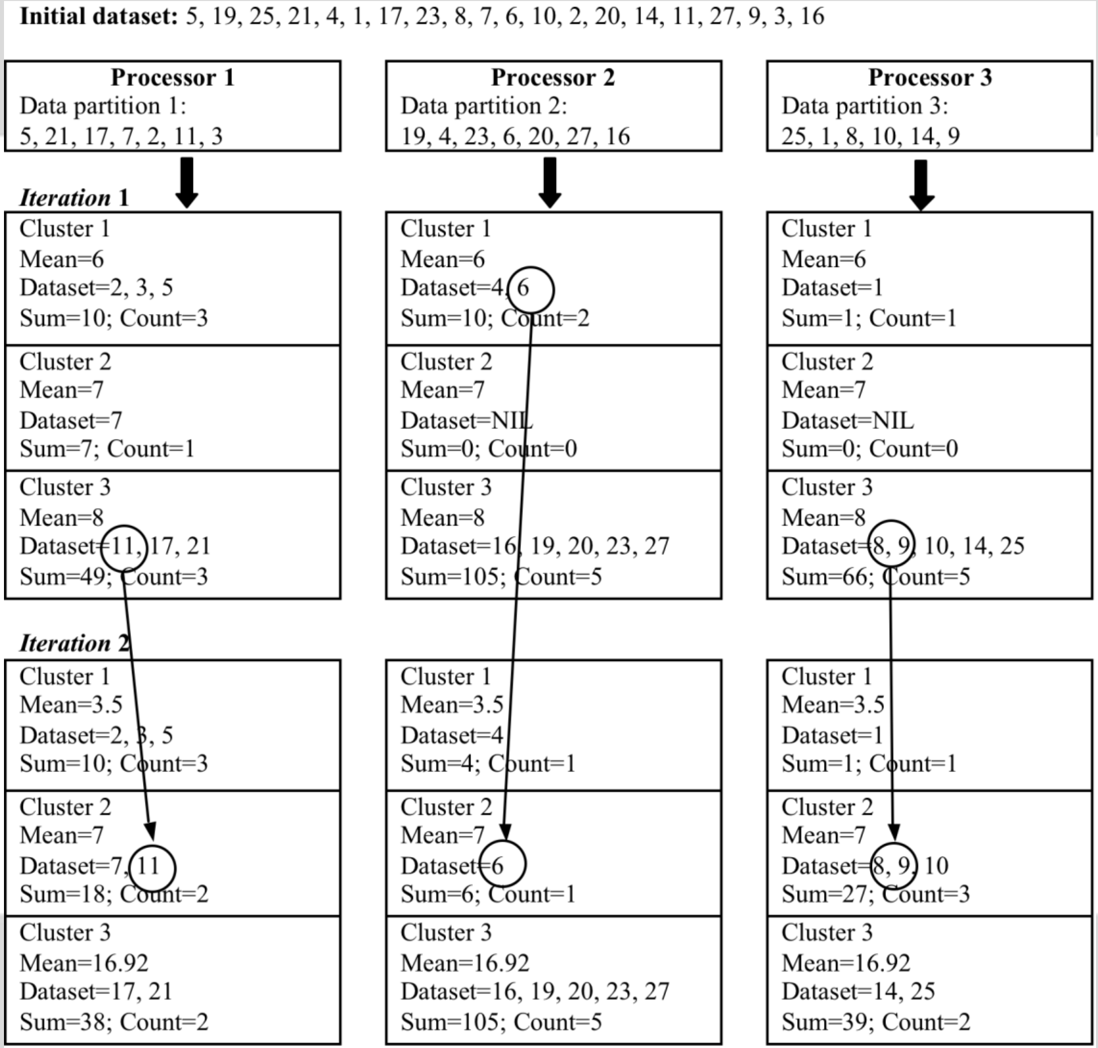
k-means

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

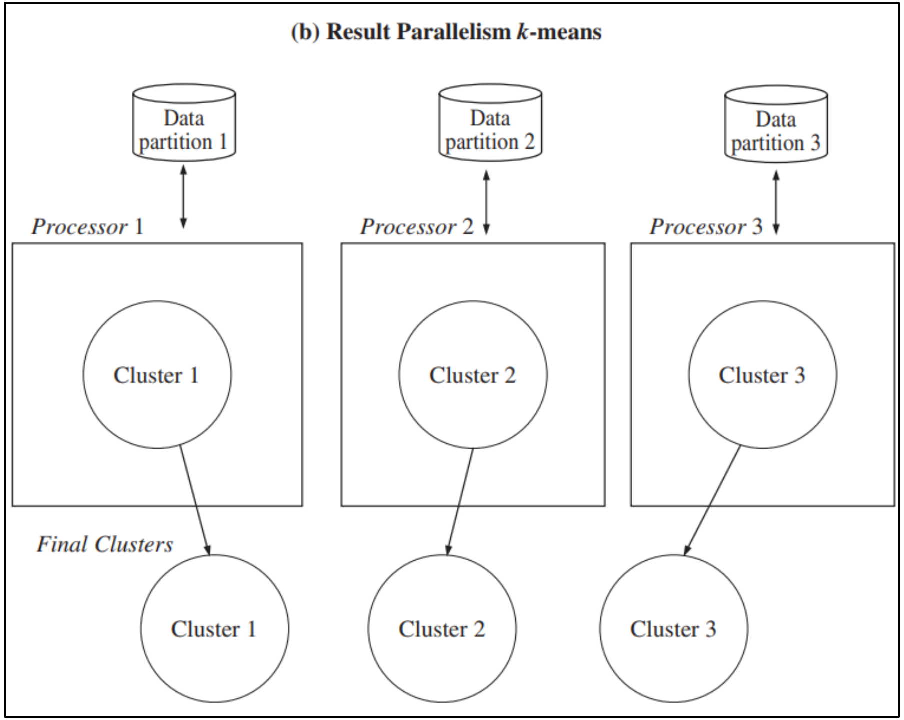
Cluster 1 = 1, 2, 3, 4, 5, 6
Cluster 2 = 7, 8, 9, 10, 11, 14
Cluster 3 = 16, 17, 19, 20, 21, 23, 25, 27



Parallel K-means clustering

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

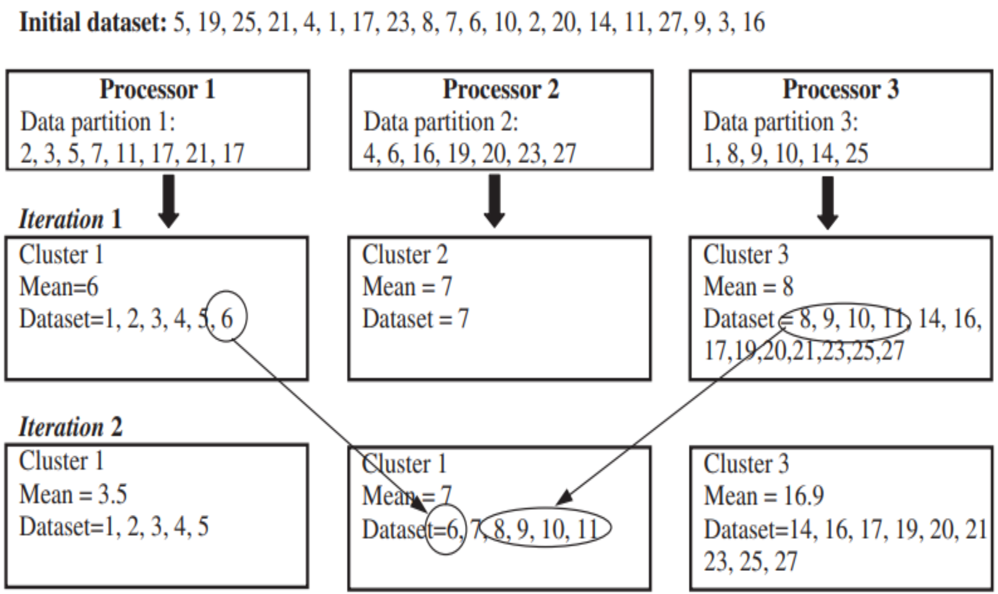
- Focuses on final/result clusters partitioning
- Each processor will work on a particular target cluster
- For example, from the very beginning, processor 1 will produce only one cluster assigned to it, that is cluster 1.
- During the iteration, the memberships of cluster can change. -→ data movement across processors



Parallel K-means

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

- Example: Data partitioning using round-robin
- Each processor is allocated only one cluster.
- Three initial means are distributed among the three processors,
- Data points may move from one processor to another at each iteration to join a cluster in a different processor
- Since a cluster is processed by one processor, calculating the mean is straightforward because all the data points within a cluster are located at the same processor

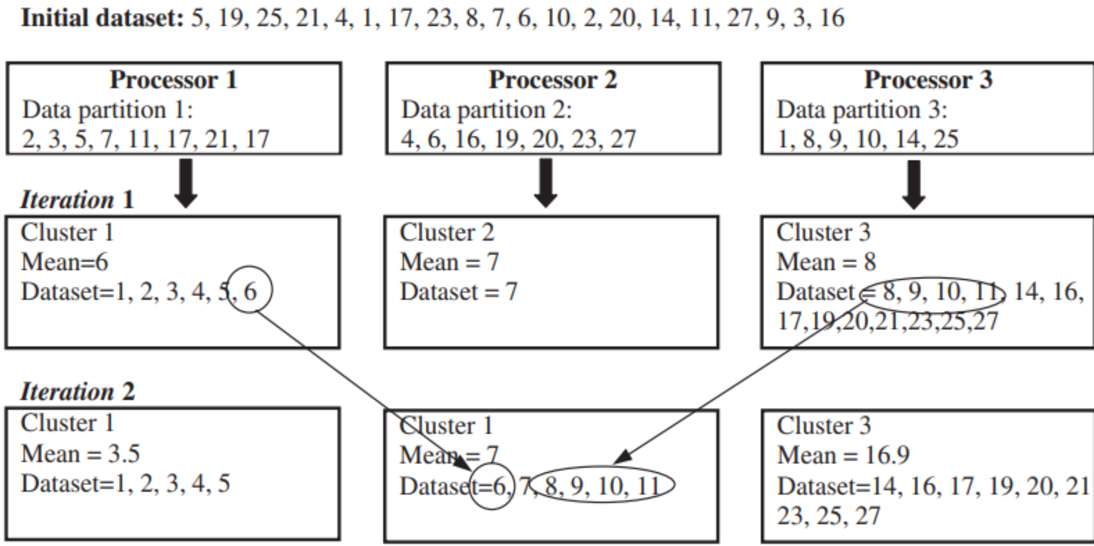


Parallel K-means

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

At the end, the final cluster result is basically the union of all local clusters from each processor.

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



Data parallelism vs Result parallelism

Data parallelism

- Parallelism is created due to the fragmentation of initial input data
- Each processor focuses on its partition of the dataset
- Final results are formed by combining all local results produced by individual processors.

Result parallelism

- Focuses on the fragmentation of the results, not necessarily the input data.
- Each processor focuses on its target result partition.