Data Analysis & Visualisation

CSC3062

BEng (CS & SE), MEng (CS & SE), BIT & CIT

Dr Reza Rafiee

Semester 1 2019

Classification vs. clustering

Objectives/samples with classes → Pre-processing → Feature selection → Find a decision boundary → Final model

It's called model training or classifier training stage

Objects/samples with no classes → Pre-processing → Feature selection → Feature space partitioning

Feature selection for high separability

Unsupervised learning

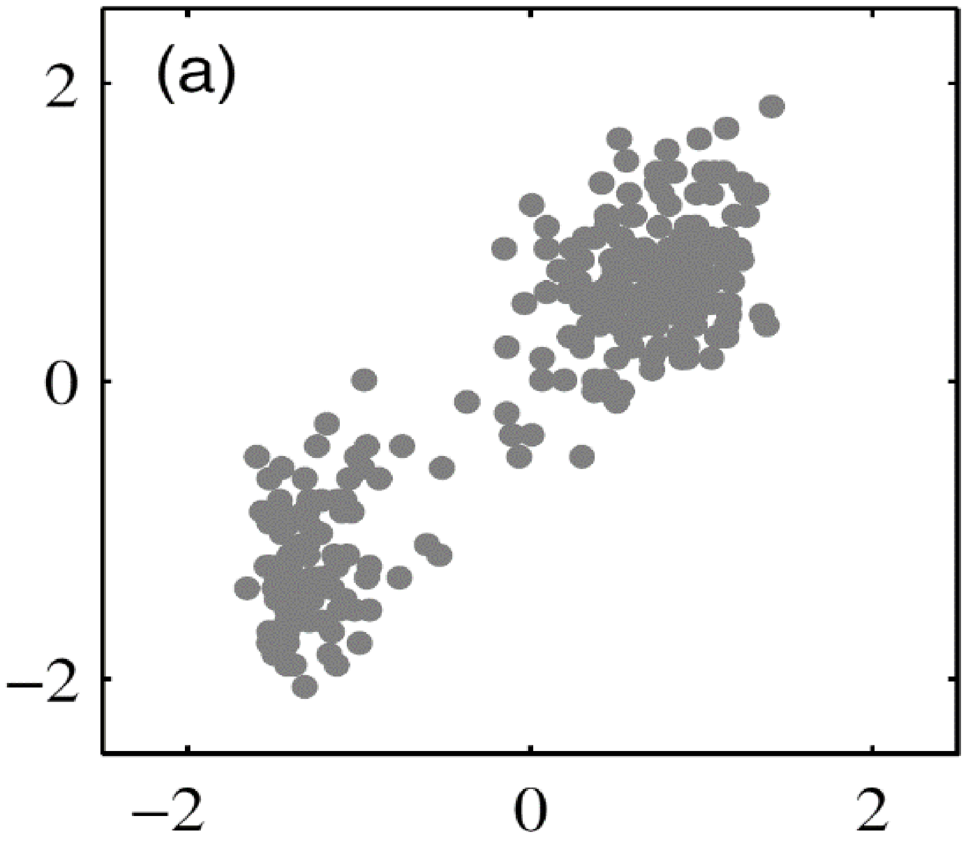
K-means clustering

- k-means is one of the simplest unsupervised learning algorithms
- It classifies a given data set through a certain number of clusters (let's say k clusters)

K-means clustering

QUEEN'S UNIVERSITY BELFAST
SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Let's cluster the following data points using k-means algorithm



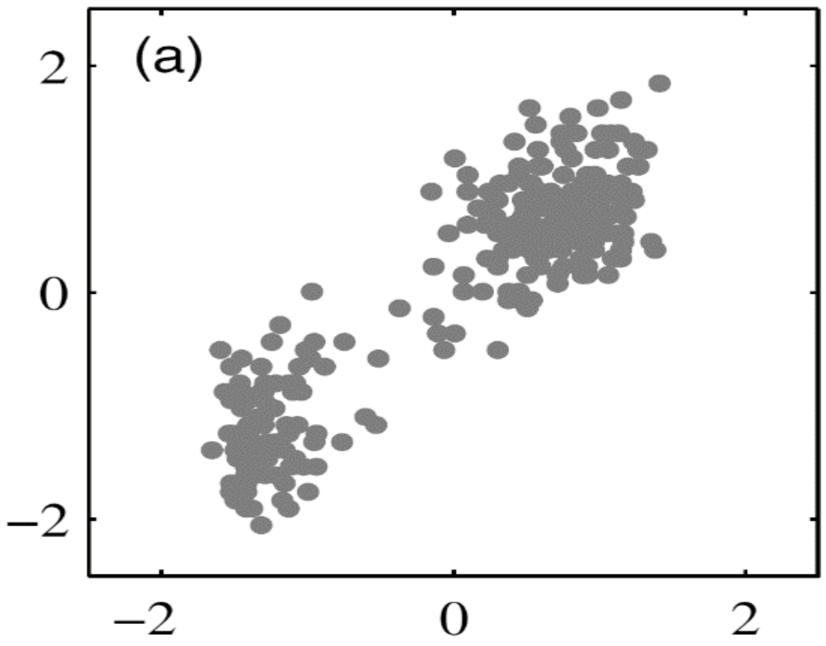
Images originated from Pattern Recognition and Machine Learning by Christopher M. Bishop

QUEEN'S UNIVERSITY BELFAST | SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

K-means clustering

Basic algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers (or cluster centroids)



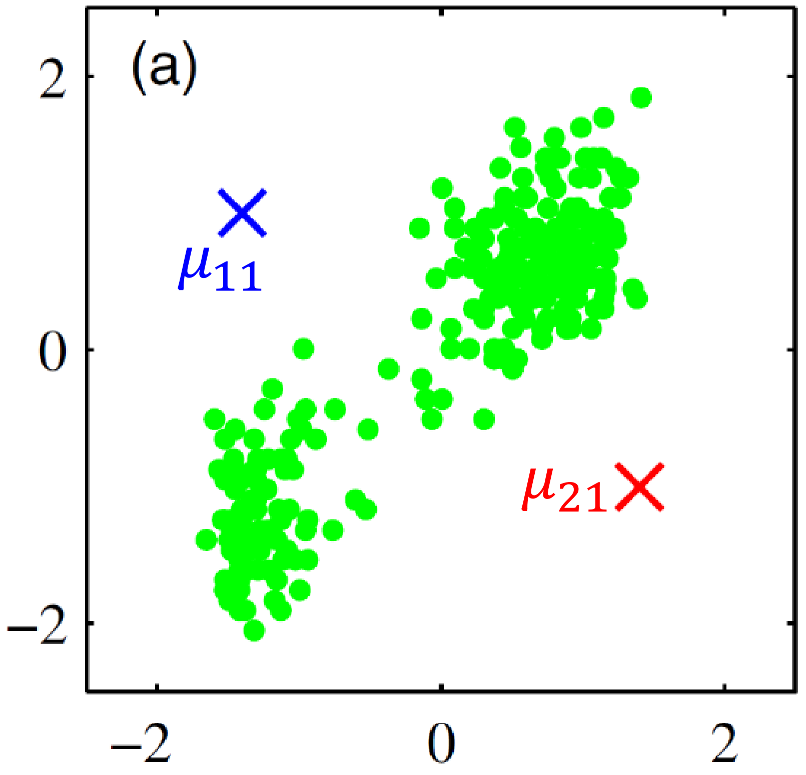
Images originated from Pattern Recognition and Machine Learning by Christopher M. Bishop

K-means clustering

Basic algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers (or cluster centroids)

k=2

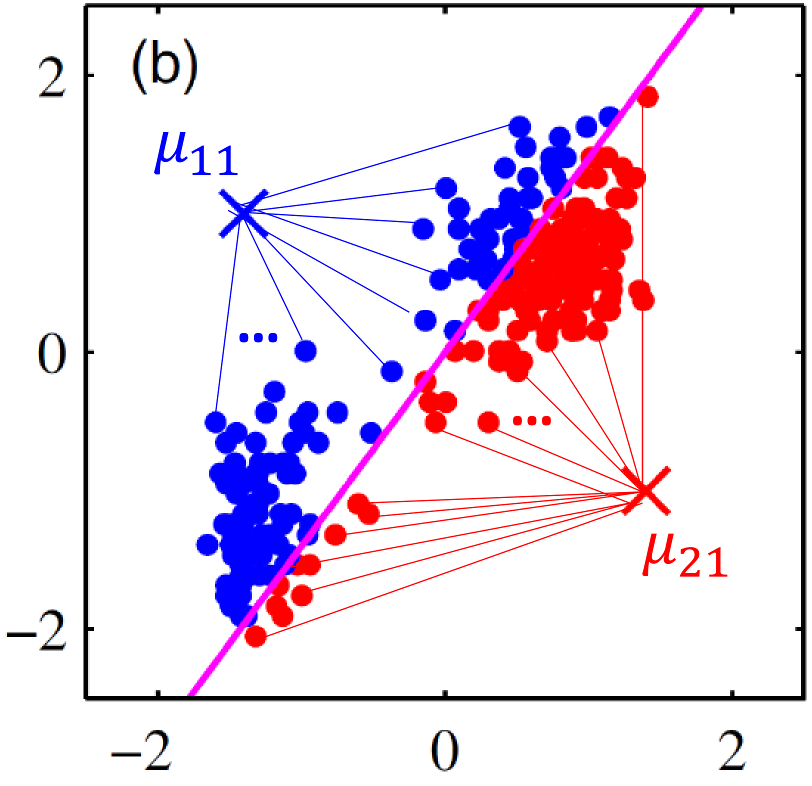


Images originated from Pattern Recognition and Machine Learning by Christopher M. Bishop

K-means clustering

Basic algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
  - What type of distance should we use? E.g., Euclidean distance
- Step 4: assign each data point to the closest cluster center (centroid)

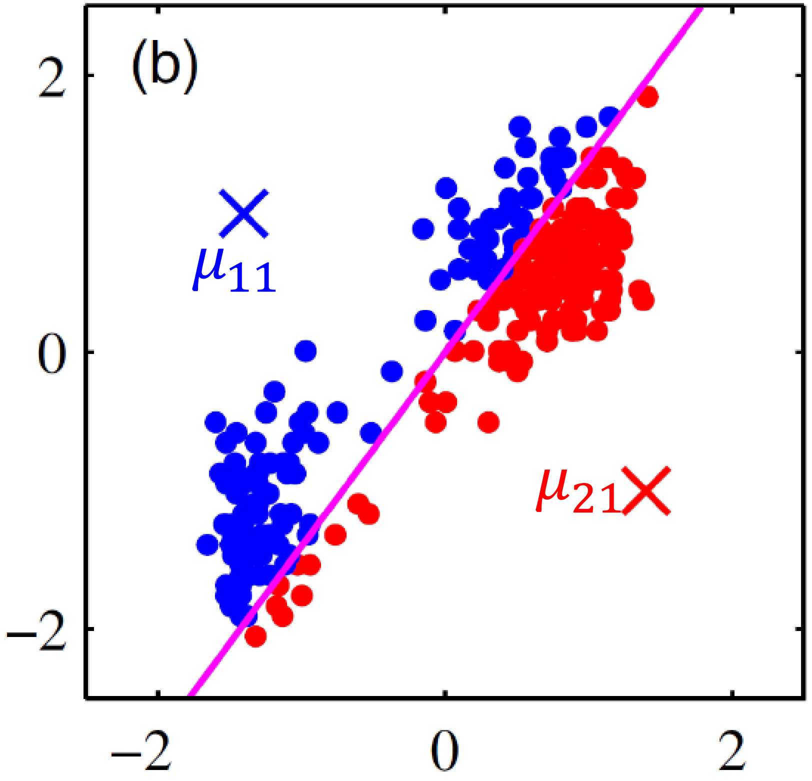


Distances partially illustrated

K-means clustering

Basic algorithm:

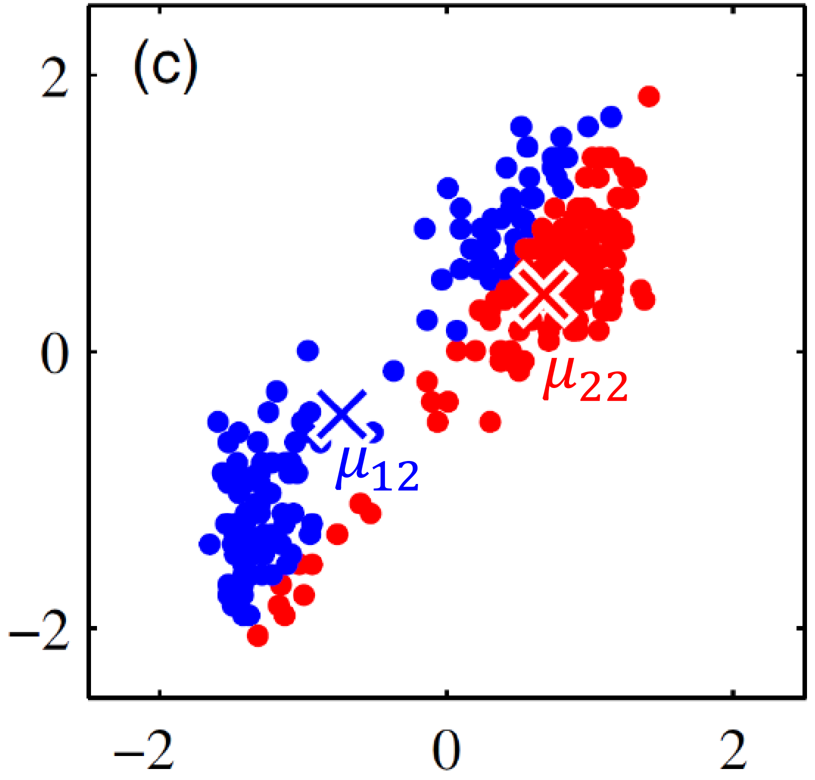
- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
  - What type of distance should we use? E.g., Euclidean distance
- Step 4: assign each data point to the closest cluster center (centroid)



K-means clustering

Basic algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
  - What type of distance should we use? E.g., Euclidean distance
- Step 4: assign each data point to the closest cluster center (centroid)
- Step 5: calculate new centroids as the mean of the data points that belong to the centroid of the previous step

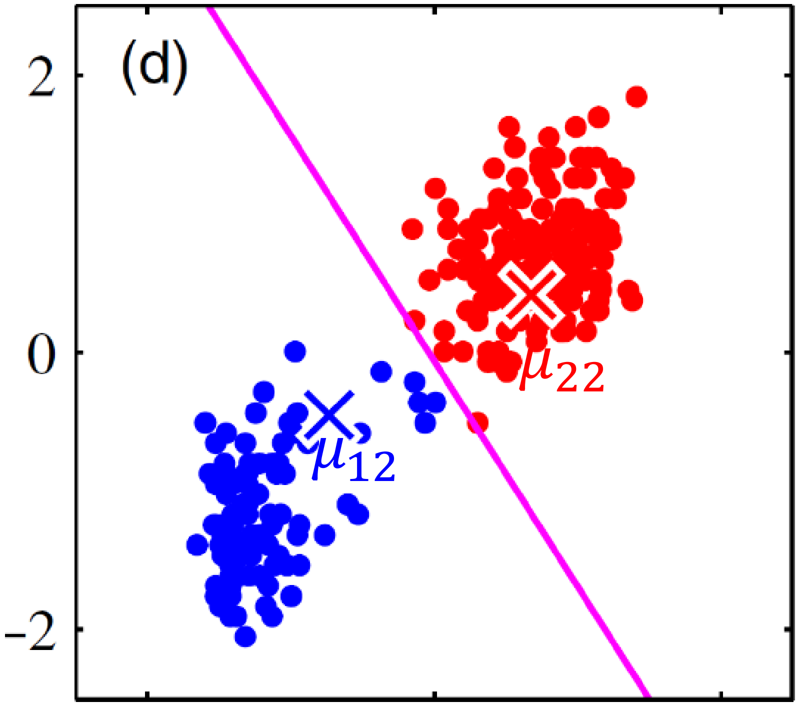


Distances partially illustrated

K-means clustering

Basic algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
  - What type of distance should we use? E.g., Euclidean distance
- Step 4: assign each data point to the closest cluster center (centroid)
- Step 5: calculate new centroids as the mean of the data points that belong to the centroid of the previous step

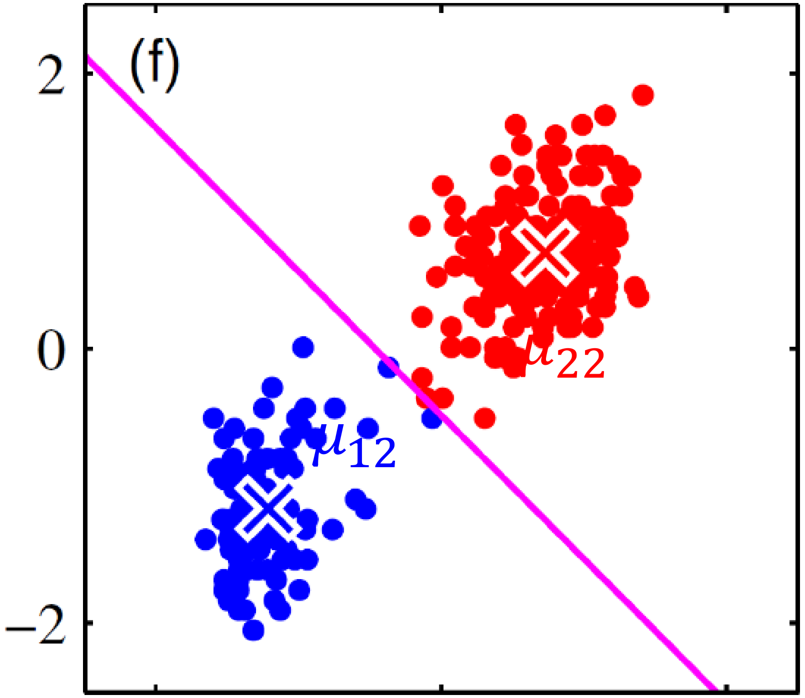


Distances partially illustrated

K-means clustering

Basic algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
  - What type of distance should we use? E.g., Euclidean distance
- Step 4: assign each data point to the closest cluster center (centroid)
- Step 5: calculate new centroids as the mean of the data points that belong to the centroid of the previous step
- Repeat Step 3-5 until a final stop condition

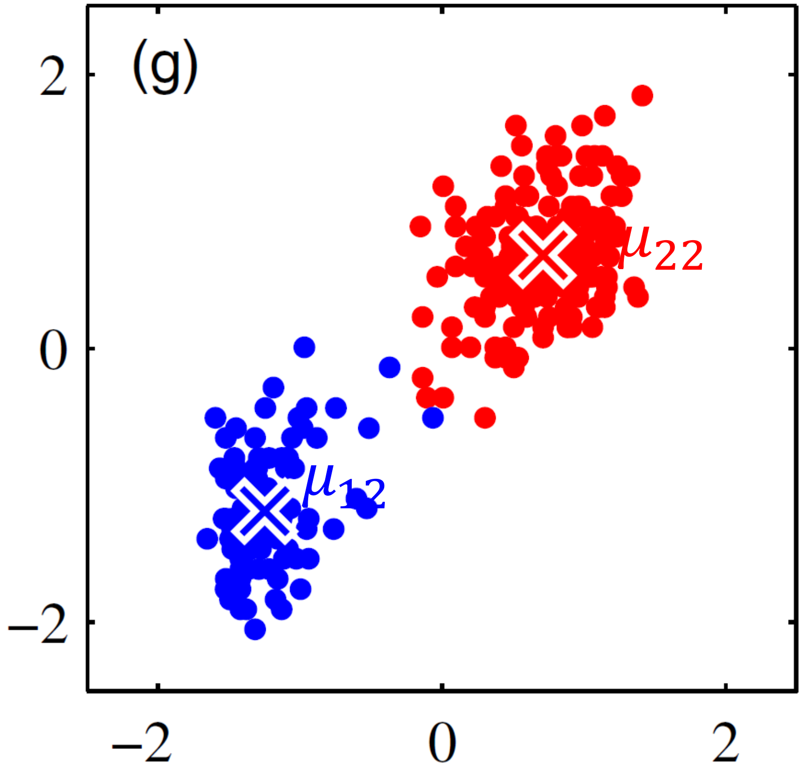


Distances partially illustrated

K-means clustering

Basic algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
    - What type of distance should we use? E.g., Euclidean distance
- Step 4: assign each data point to the closest cluster center (centroid)
- Step 5: calculate new centroids as the mean of the data points that belong to the centroid of the previous step
- Repeat Step 3-5 until a final stop condition

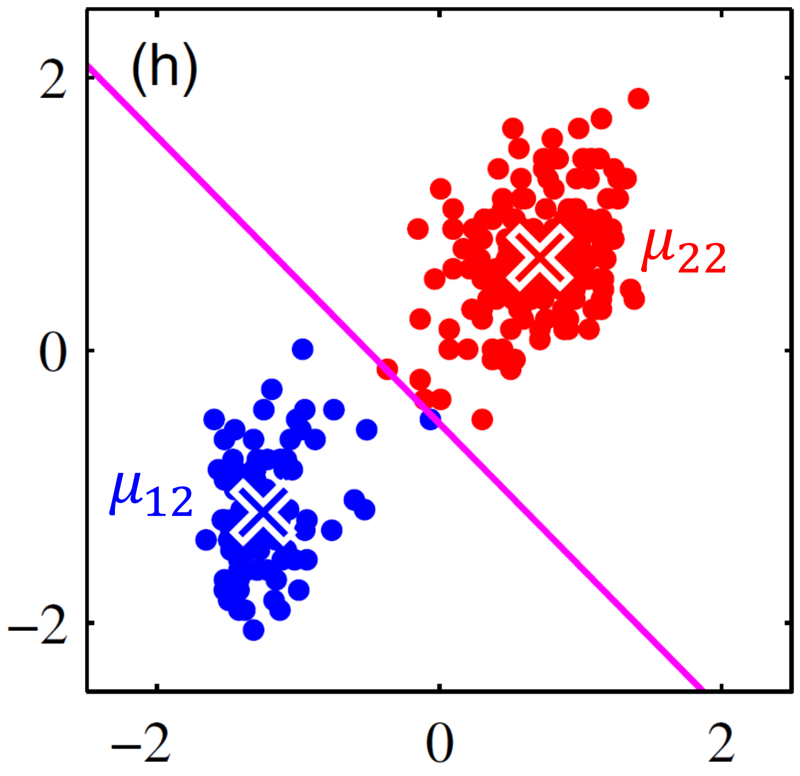


Distances partially illustrated

K-means clustering

Basic algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
  - What type of distance should we use? E.g., Euclidean distance
- Step 4: assign each data point to the closest cluster center (centroid)
- Step 5: calculate new centroids as the mean of the data points that belong to the centroid of the previous step
- Repeat Step 3-5 until a final stop condition

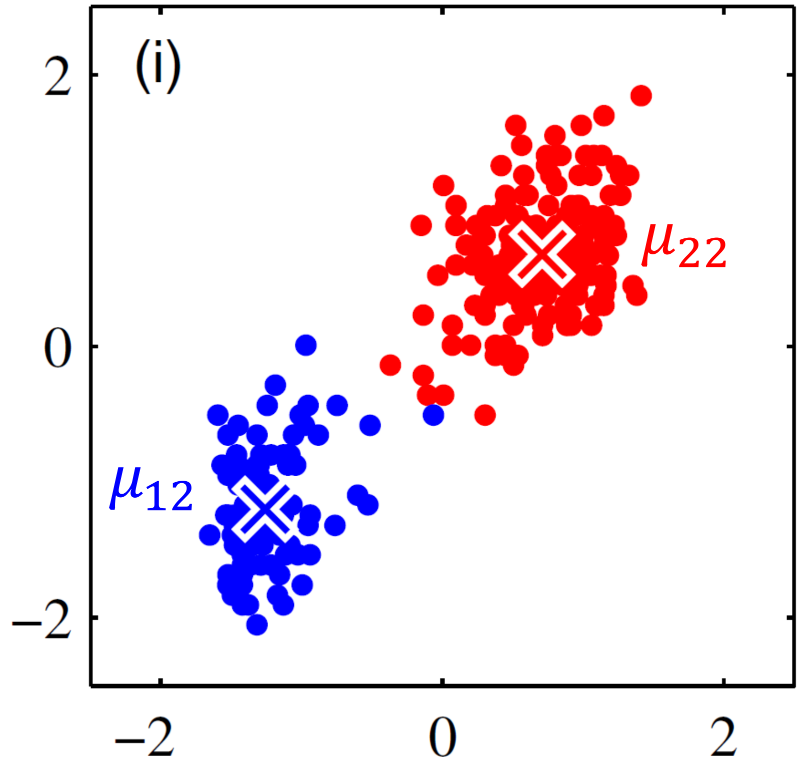


Distances partially illustrated

K-means clustering

Basic algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
  - What type of distance should we use? E.g., Euclidean distance
- Step 4: assign each data point to the closest cluster center (centroid)
- Step 5: calculate new centroids as the mean of the data points that belong to the centroid of the previous step
- Repeat Step 3-5 until a final stop condition

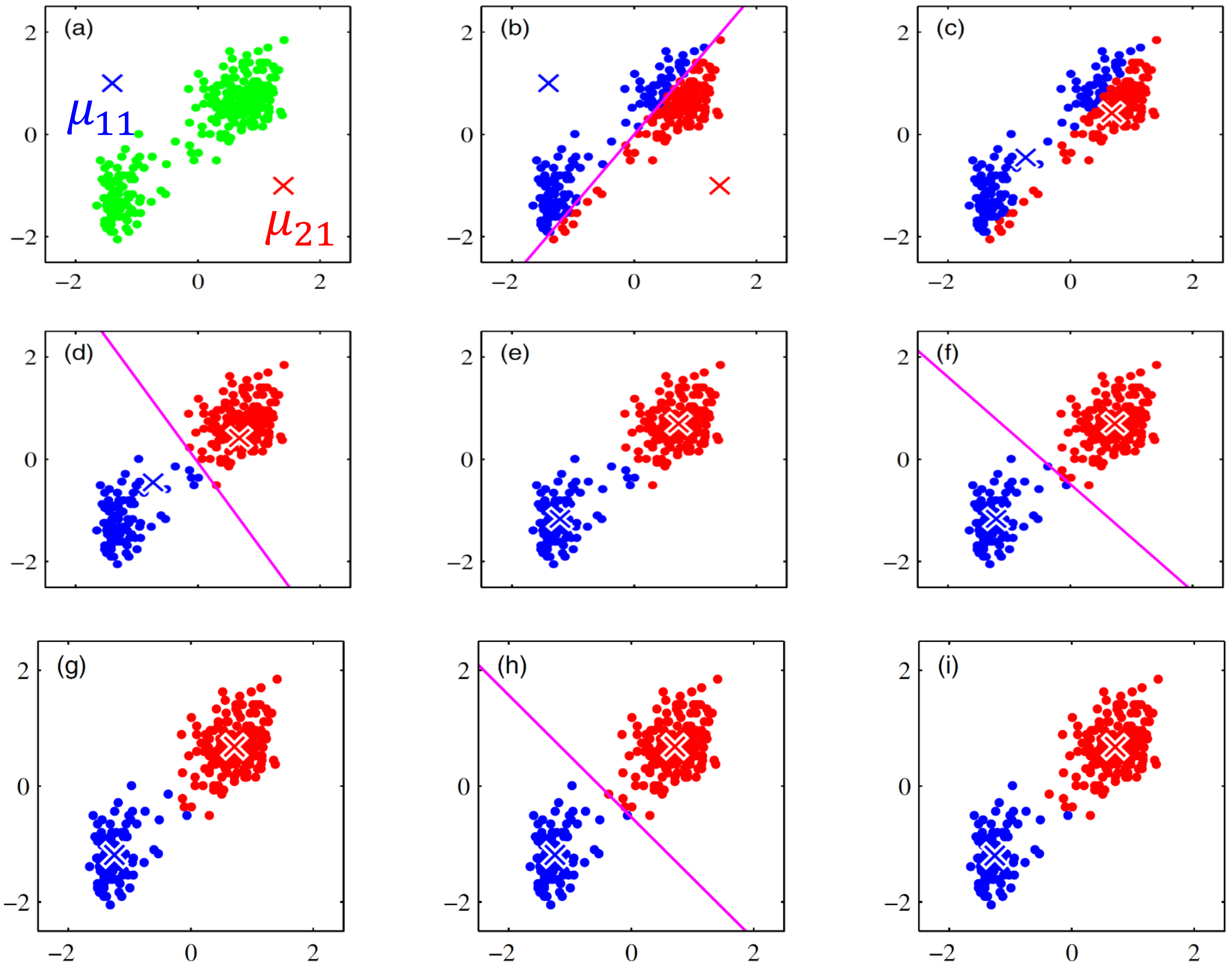


Distances partially illustrated

K-means clustering

Initial cluster centres: μ_1 μ_2

Illustration of *k*-means algorithm
(a) Green points denote the data set in a two-dimensional Euclidean space



Images originated from Pattern Recognition and Machine Learning by Christopher M. Bishop

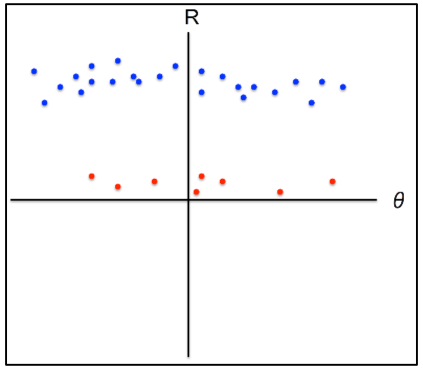
K-means clustering

Basic algorithm:

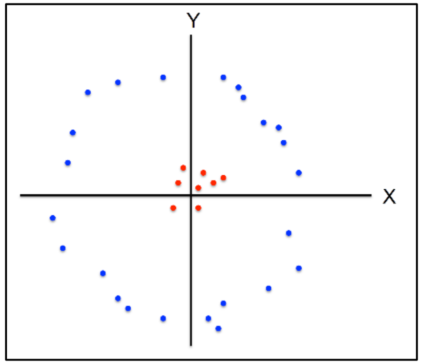
- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
  - What type of distance should we use? E.g., Euclidean distance
- <u>Step 4: assign each data point to the closest cluster center (centroid)</u>
- Step 5: calculate new centroids as the mean of the data points that belong to the centroid of the previous step
- Repeat Step 3-5 until a final stop condition (if no data point was reassigned then stop).

K-means clustering

- Strengths
  - Simple & fast and can be applied to high-dimensional large data
  - Finds cluster centres that minimize conditional variance (good representation of data)
  - Easy to implement
- Weaknesses
  - Need to choose k
  - Sensitive to outliers
  - Prone to local minima and no guarantee of optimal solution (local optima)
    - Repeat with different starting values
  - Difficult to guess the correct “k”



Changing features & distance function



K-means algorithm is not able to properly cluster this data points

Some practical tips for clustering

Assume, we are given a dataset for the purpose of clustering analysis

How to choose a reliable clustering technique for your dataset?

Some practical tips for clustering

Assume, we are given a dataset for the purpose of clustering analysis

How to choose a reliable clustering technique for your dataset?

- **Evaluate your dataset from different aspects**
  - What type of features (e.g., numeric or categorical)?
  - The size of the dataset (e.g., large or small)
  - Number of feature (i.e., attributes), Is it a high-dimensional dataset?
  - Assessing outliers and missing
- **Consider consensus clustering**
- **Evaluate the reliability (i.e., consistency/robust) of the clustering result**

Some practical tips for clustering

Assume, we are given a dataset for the purpose of clustering analysis

How to choose a reliable clustering technique for your dataset?

- **Evaluate your dataset from different aspects**
  - What type of features (e.g., numeric or categorical)?
  - The size of the dataset (e.g., large or small)
  - Number of feature (i.e., attributes), Is it a high-dimensional dataset?
  - Assessing outliers and missing
- **Consider consensus clustering**
- **Evaluate the reliability (i.e., consistency/robust) of the clustering result**

Some practical tips for clustering

Assume, we are given a dataset for the purpose of clustering analysis

How to choose a reliable clustering technique for your dataset?

- **Evaluate your dataset from different aspects**
  - What type of features (e.g., numeric or categorical)?
  - The size of the dataset (e.g., large or small)
  - Number of feature (i.e., attributes), Is it a high-dimensional dataset?
  - Assessing outliers and missing
- **Consider consensus clustering**
- **Evaluate the reliability (i.e., consistency/robust) of the clustering result**

Consensus clustering

Assume, we are given a dataset for the purpose of clustering analysis

1) No knowledge of about the number of clusters
2) Clustering methods are sensitive to initialisation settings
3) The lack of a reliable validation technique when using clustering
   a) We need a measure of confidence for cluster numbers and cluster assignment

Consensus clustering[1]

Assume, we are given a dataset for the purpose of clustering analysis

1) Multiple runs of a clustering algorithm
   a) Determine the number of clusters and assess the stability of the discovered clusters
   b) In k-means clustering: with using random restart

2) Aggregating the cluster (label) results of different clustering algorithms

[1] Ensemble clustering

Consensus clustering[1]

Assume, we are given a dataset for the purpose of clustering analysis

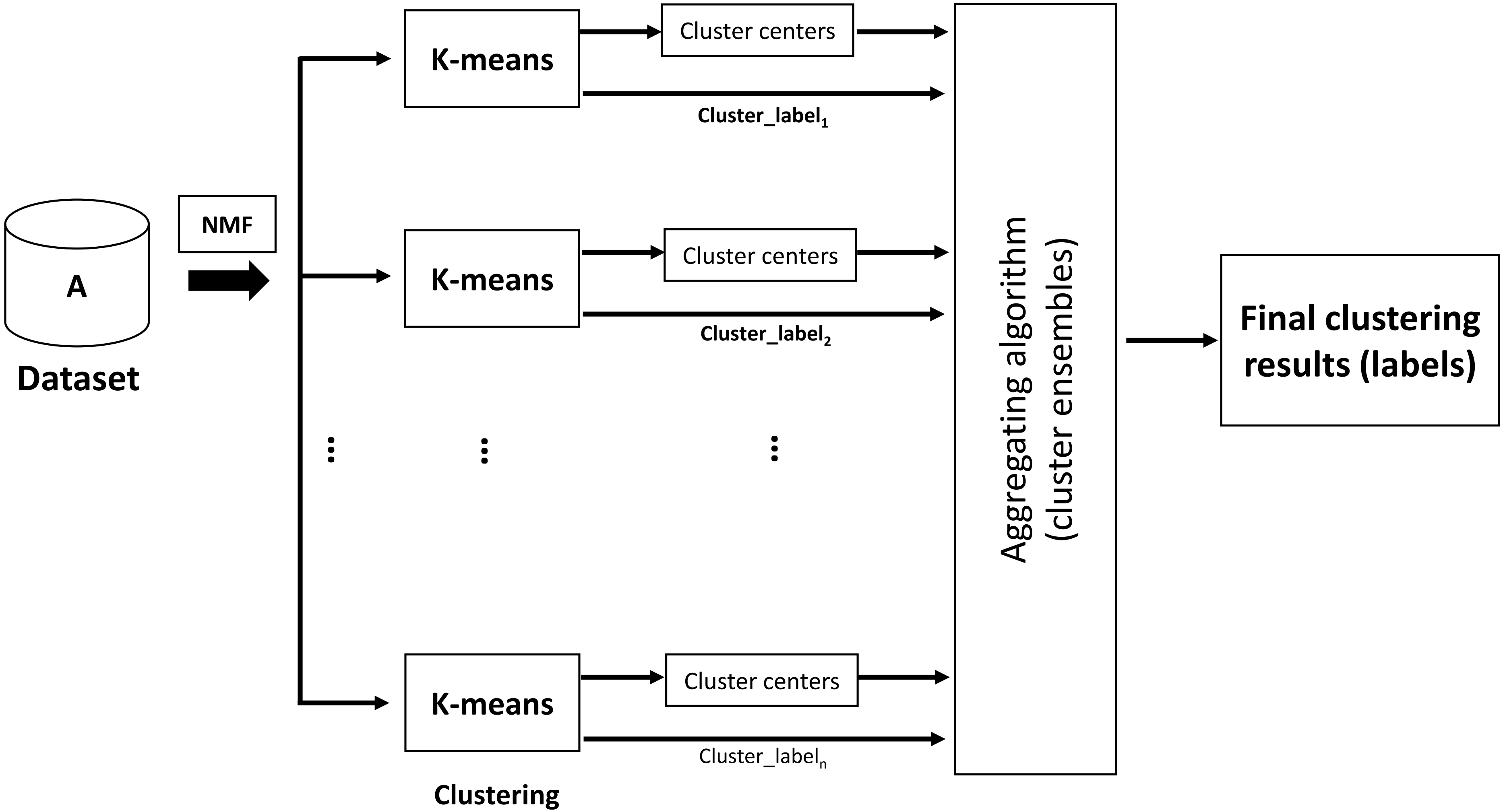
1) Multiple runs of a clustering algorithm
   a) Determine the number of clusters and assess the stability of the discovered clusters
   b) In k-means clustering: with using random restart

2) Aggregating the cluster (label) results of different clustering algorithms

[1] Ensemble clustering

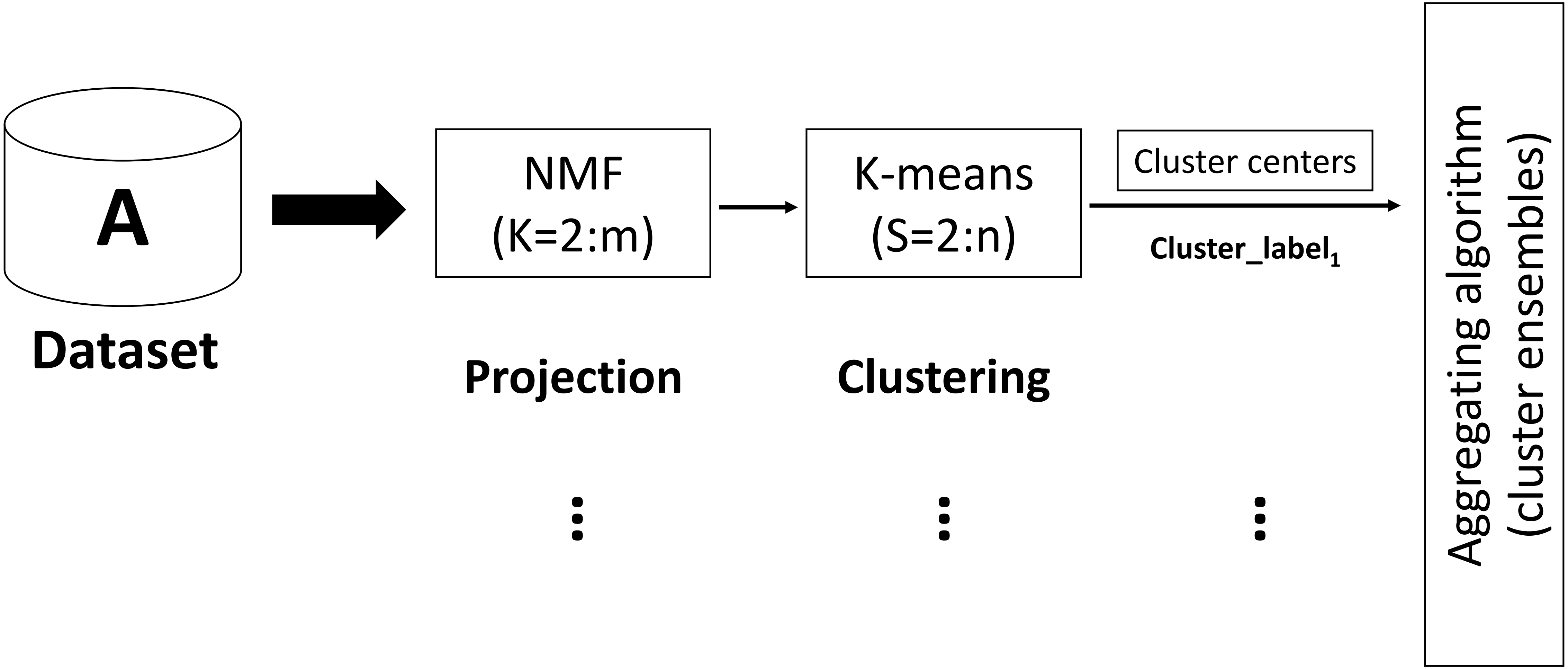
Consensus approach

1) Multiple runs of a clustering algorithm



Consensus approach

A comprehensive Ensemble approach for unsupervised clustering using NMF projection and k-means clustering



The value of m is dependent on the number of input features

n=12 samples with 4 subgroups

After running NMF on our input dataset 17×220 ===> k=4, H matrix

Consider only 12 samples of H matrix (for the sake of simplicity)

```
#-----------------------------------------------------------------------------------------------
# Consider only 12 samples out of 220 with 4 metagenes
Small_dataset_cluster_analysis <- read.csv("H_matrix_17_8_k4_4.csv",row.names = 1)
rownames(Small_dataset_cluster_analysis) <- c("Metagene_1","Metagene_2","Metagene_3","Metagene_4")
min(Small_dataset_cluster_analysis)  # [1] 4.14e-70
max(Small_dataset_cluster_analysis)  # [1] 9.434869
Small_dataset_cluster_analysis_0To1 <- Data_Range_Into_01(Small_dataset_cluster_analysis)
min(Small_dataset_cluster_analysis_0To1)
max(Small_dataset_cluster_analysis_0To1)
```

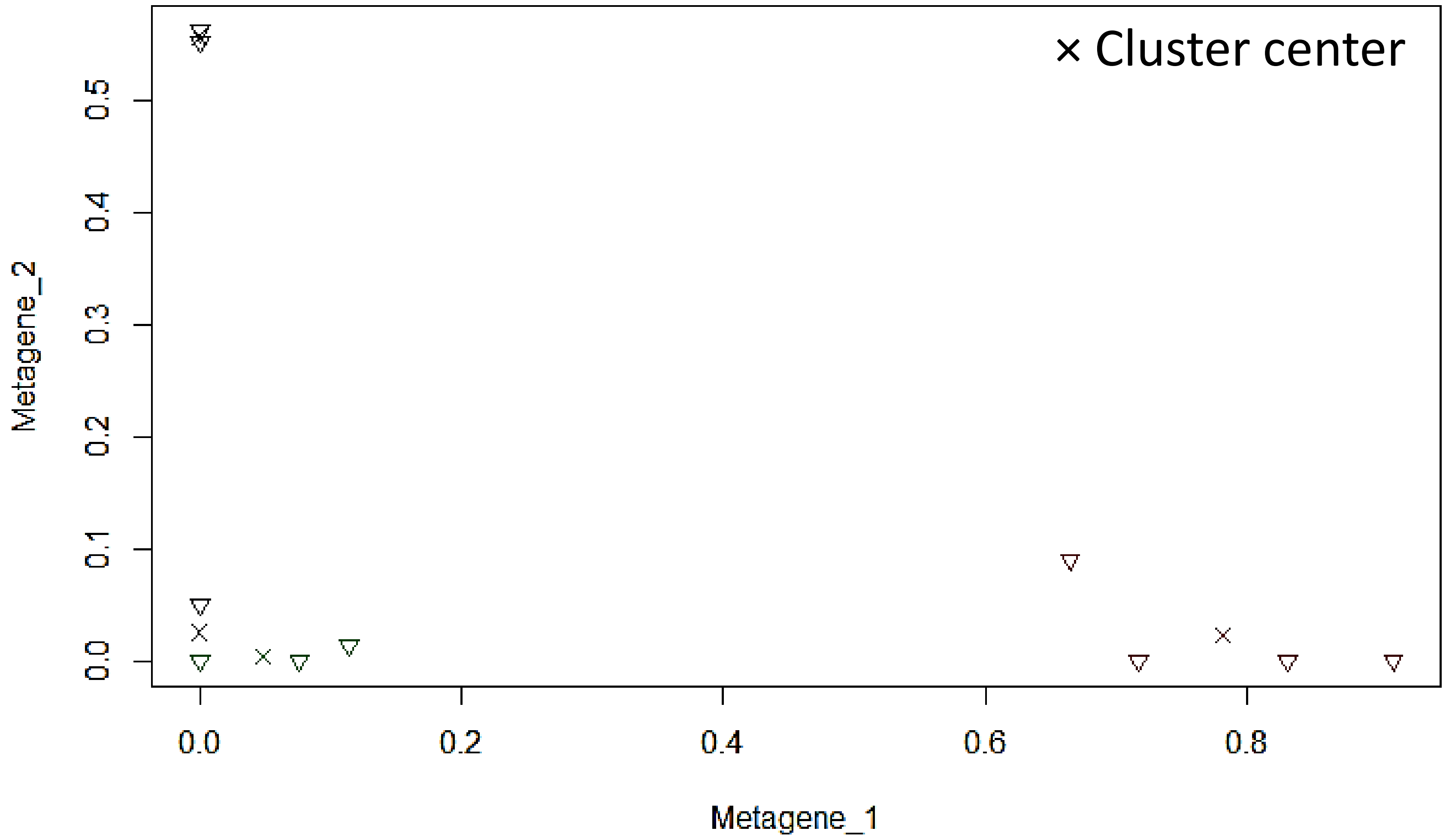
n=12 samples with 4 subgroups

	CSC3062_108_2	CSC3062_109_4	CSC3062_110_4	CSC3062_112_2	CSC3062_783_3	CSC3062_145_3
Metagene_1	1.145277e-01	1.916895e-50	2.654951e-40	7.633172e-02	3.608274e-32	7.042284e-28
Metagene_2	1.338042e-02	5.529235e-01	5.625382e-01	4.172066e-27	5.022959e-02	1.881889e-05
Metagene_3	5.842943e-19	5.115138e-43	1.629874e-28	2.634450e-34	6.117725e-01	6.623634e-01
Metagene_4	9.603256e-01	2.808713e-27	4.787113e-29	9.671474e-01	1.660626e-34	5.350906e-39

	CSC3062_649_1	CSC3062_115_1	CSC3062_670_2	CSC3062_50080_1	CSC3062_436_1	CSC3062_674_2
Metagene_1	7.176776e-01	9.121094e-01	2.142412e-28	8.314318e-01	6.650897e-01	1.424858e-17
Metagene_2	0.000000e+00	1.312099e-40	2.695954e-17	1.158338e-18	8.997966e-02	3.280249e-12
Metagene_3	1.759033e-70	3.300750e-21	3.208493e-17	1.691378e-40	3.382756e-17	2.059872e-02
Metagene_4	6.929525e-63	3.516017e-59	9.679785e-01	4.684605e-20	1.916895e-23	1.000000e+00

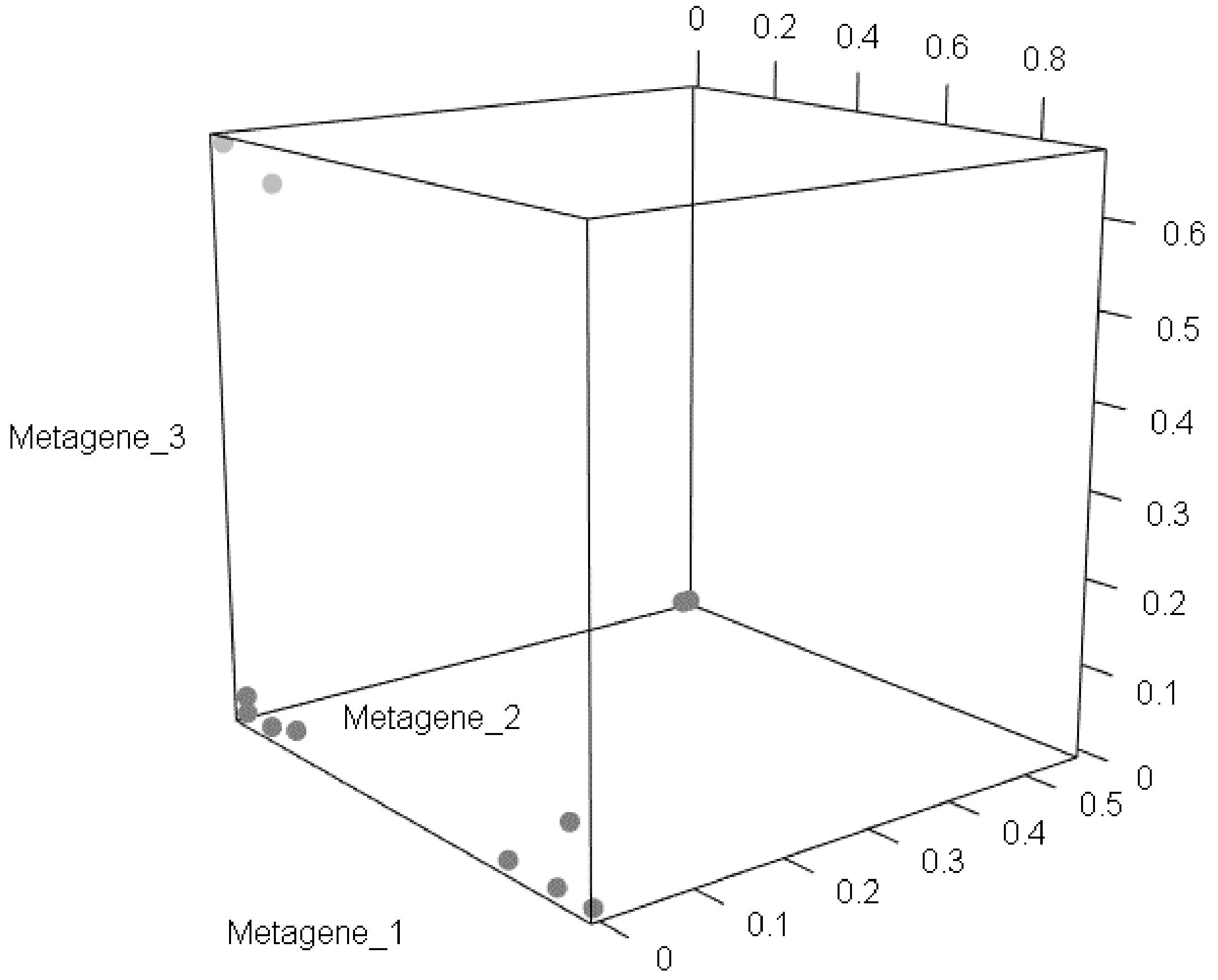
After running k-means on these 12 samples

```
K_means_Model <- kmeans(t(Small_dataset_cluster_analysis_0To1),centers = 4, iter.max = 50,nstart = 5) #
```



QUEEN'S UNIVERSITY BELFAST
SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Using plot3d() to visualise samples



Visualising the H matrix using PCA

Several runs of k-means

```
# Creating a matrix of all labels of different k-means runnings

Matrix_labels_different_runs <- matrix(nrow = ncol(Small_dataset_cluster_analysis_0To1), ncol = 10,0)
rownames(Matrix_labels_different_runs) <- colnames(Small_dataset_cluster_analysis_0To1)
for (j in 1:10) {
    K_means_Model <- kmeans(t(Small_dataset_cluster_analysis_0To1),centers = 4, iter.max = 50,nstart = 5) #
    #trying several random starts (nstart> 1) is often recommended.
    Matrix_labels_different_runs[,j] <- K_means_Model$cluster
} # for


plot(t(Small_dataset_cluster_analysis_0To1), col = K_means_Model$cluster,pch=6)
points(K_means_Model$centers, col = 1:3, pch = 4, cex = 1)
```

Several runs of k-means

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
CSC3062_108_2	4	4	4	1	3	1	3	3	4	4
CSC3062_109_4	1	1	1	2	2	2	4	1	1	2
CSC3062_110_4	1	1	1	2	2	2	4	1	1	2
CSC3062_112_2	4	4	4	1	3	1	3	3	4	4
CSC3062_783_3	2	3	3	3	4	3	1	1	3	2
CSC3062_145_3	2	3	3	3	4	3	1	1	3	2
CSC3062_649_1	3	2	2	4	1	4	2	4	2	1
CSC3062_115_1	3	2	2	4	1	4	2	2	2	3
CSC3062_670_2	4	4	4	1	3	1	3	3	4	4
CSC3062_50080_1	3	2	2	4	1	4	2	2	2	3
CSC3062_436_1	3	2	2	4	1	4	2	4	2	1
CSC3062_674_2	4	4	4	1	3	1	3	3	4	4

Different cluster labels obtained from several runs (m=10) of k-means clustering algorithms

Cluster labels (V_1, V_2, …, V_{10}) are not unique!

Inspection of the label vectors

$V_1 = (4,1,1,4,2,2,3,3,4,3,3,4)$

$V_2 = (4,1,1,4,3,3,2,2,4,2,2,4)$

$V_3 = (4,1,1,4,3,3,2,2,4,2,2,4)$

$V_4 = (1,2,2,1,3,3,4,4,1,4,4,1)$

$V_5 = (3,2,2,3,4,4,1,1,3,1,1,3)$

$V_6 = (1,2,2,1,3,3,4,4,1,4,4,1)$

$V_7 = (3,4,4,3,1,1,2,2,3,2,2,3)$

$V_8 = (3,1,1,3,1,1,4,2,3,2,4,3)$

$V_9 = (4,1,1,4,3,3,2,2,4,2,2,4)$

$V_{10} = (4,2,2,4,2,2,1,3,4,3,1,4)$

Which clustering algorithms are creating same/similar cluster labels?

Which samples are not confidently clustered?

Different cluster labels obtained from several runs (m=10) of k-means clustering algorithms

Cluster labels (V_1, V_2, …, V_{10}) are **not unique**

QUEEN'S UNIVERSITY BELFAST | SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Inspection of the label vectors

$V_1 = (4,1,1,4,2,2,3,3,4,3,3,4)$

$V_2 = (4,1,1,4,3,3,2,2,4,2,2,4)$

$V_3 = (4,1,1,4,3,3,2,2,4,2,2,4)$

$V_4 = (1,2,2,1,3,3,4,4,1,4,4,1)$

$V_5 = (3,2,2,3,4,4,1,1,3,1,1,3)$

$V_6 = (1,2,2,1,3,3,4,4,1,4,4,1)$

$V_7 = (3,4,4,3,1,1,2,2,3,2,2,3)$

$V_8 = (3,1,1,3,1,1,4,2,3,2,4,3)$

$V_9 = (4,1,1,4,3,3,2,2,4,2,2,4)$

$V_{10} = (4,2,2,4,2,2,1,3,4,3,1,4)$

Clusterings V2 and V3 are identical.

Different cluster labels obtained from several runs (m=10) of k-means clustering algorithms
Cluster labels (V_1, V_2, …, V_{10}) are **not unique**

QUEEN'S UNIVERSITY BELFAST

SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Inspection of the label vectors

$$V_1 = (4,1,1,4,2,2,3,3,4,3,3,4)$$

$$V_2 = (4,1,1,4,3,3,2,2,4,2,2,4)$$

$$V_3 = (4,1,1,4,3,3,2,2,4,2,2,4)$$

$$V_4 = (1,2,2,1,3,3,4,4,1,4,4,1)$$

$$V_5 = (3,2,2,3,4,4,1,1,3,1,1,3)$$

$$V_6 = (1,2,2,1,3,3,4,4,1,4,4,1)$$

$$V_7 = (3,4,4,3,1,1,2,2,3,2,2,3)$$

$$V_8 = (3,1,1,3,1,1,4,2,3,2,4,3)$$

$$V_9 = (4,1,1,4,3,3,2,2,4,2,2,4)$$

$$V_{10} = (4,2,2,4,2,2,1,3,4,3,1,4)$$

Clusterings V1 and V2 are logically identical.

Different cluster labels obtained from several runs (m=10) of k-means clustering algorithms

Cluster labels (V_1, V_2, …, V_{10}) are **not unique**

Inspection of the label vectors

$V_1 = (4,1,1,4,2,2,3,3,4,3,3,4)$

$V_2 = (4,1,1,4,3,3,2,2,4,2,2,4)$

$V_3 = (4,1,1,4,3,3,2,2,4,2,2,4)$

$V_4 = (1,2,2,1,3,3,4,4,1,4,4,1)$

$V_5 = (3,2,2,3,4,4,1,1,3,1,1,3)$

$V_6 = (1,2,2,1,3,3,4,4,1,4,4,1)$

$V_7 = (3,4,4,3,1,1,2,2,3,2,2,3)$

$V_8 = (3,1,1,3,1,1,4,2,3,2,4,3)$

$V_9 = (4,1,1,4,3,3,2,2,4,2,2,4)$

$V_{10} = (4,2,2,4,2,2,1,3,4,3,1,4)$

Clusterings V1, V2, V4, V5, V6, V7 and V9 are logically identical.

Different cluster labels obtained from several runs (m=10) of k-means clustering algorithms
Cluster labels (V_1, V_2, …, V_{10}) are **not unique**

Inspection of the label vectors

$$V_1 = (4,1,1,4,2,2,3,3,4,3,3,4)$$

$$V_2 = (4,1,1,4,3,3,2,2,4,2,2,4)$$

$$V_3 = (4,1,1,4,3,3,2,2,4,2,2,4)$$

$$V_4 = (1,2,2,1,3,3,4,4,1,4,4,1)$$

$$V_5 = (3,2,2,3,4,4,1,1,3,1,1,3)$$

$$V_6 = (1,2,2,1,3,3,4,4,1,4,4,1)$$

$$V_7 = (3,4,4,3,1,1,2,2,3,2,2,3)$$

$$V_8 = (3,1,1,3,1,1,4,2,3,2,4,3)$$

$$V_9 = (4,1,1,4,3,3,2,2,4,2,2,4)$$

$$V_{10} = (4,2,2,4,2,2,1,3,4,3,1,4)$$

What about V8 and V10?

Different cluster labels obtained from several runs (m=10) of k-means clustering algorithms
Cluster labels (V_1, V_2, …, V_{10}) are **not unique**

Inspection of the label vectors

$$V_2 = (4,1,1,4,3,3,2,2,4,2,2,4)$$

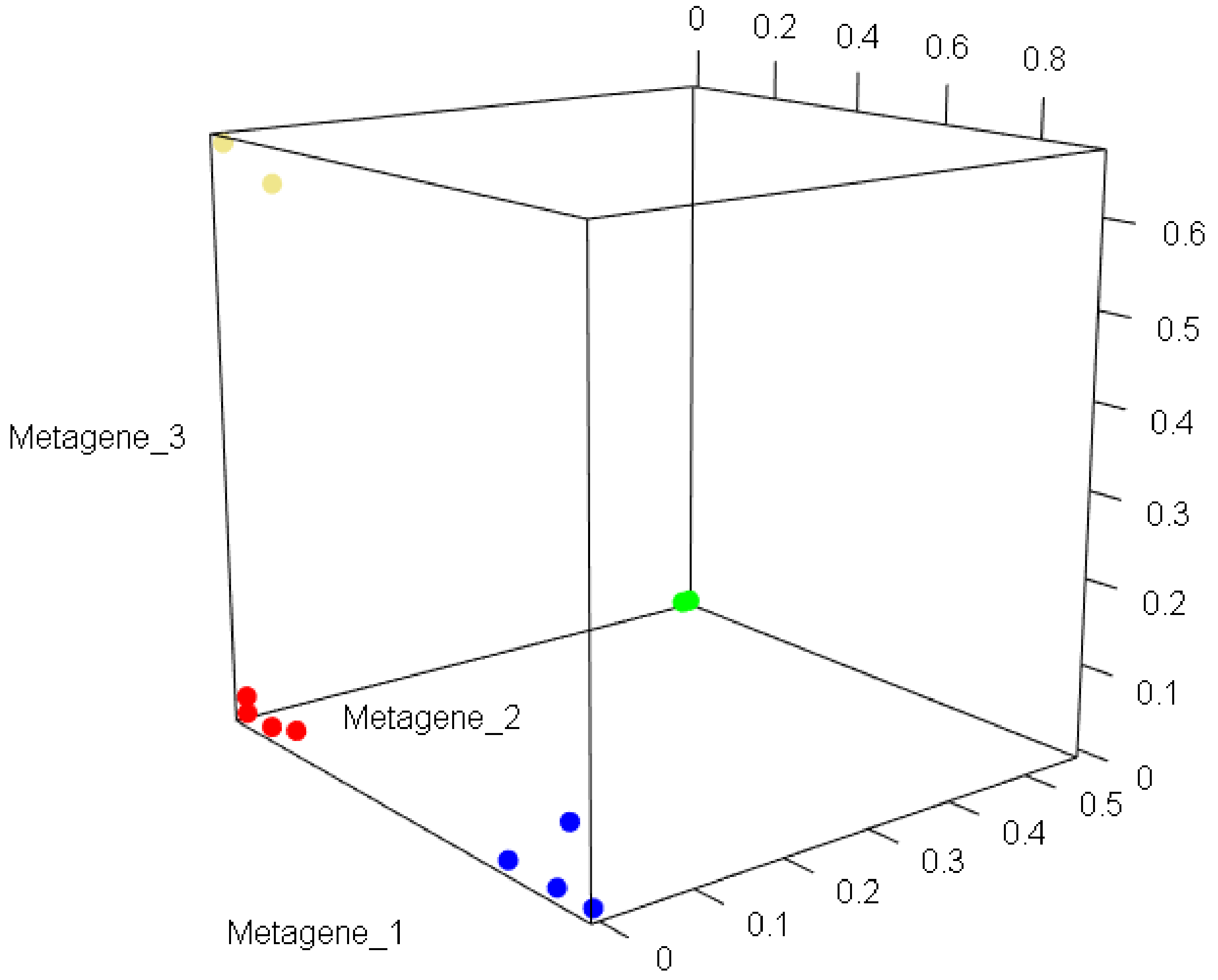
$$V_8 = (3,1,1,3,1,1,4,2,3,2,4,3)$$

$$V_{10} = (4,2,2,4,2,2,1,3,4,3,1,4)$$

What about V8 and V10?

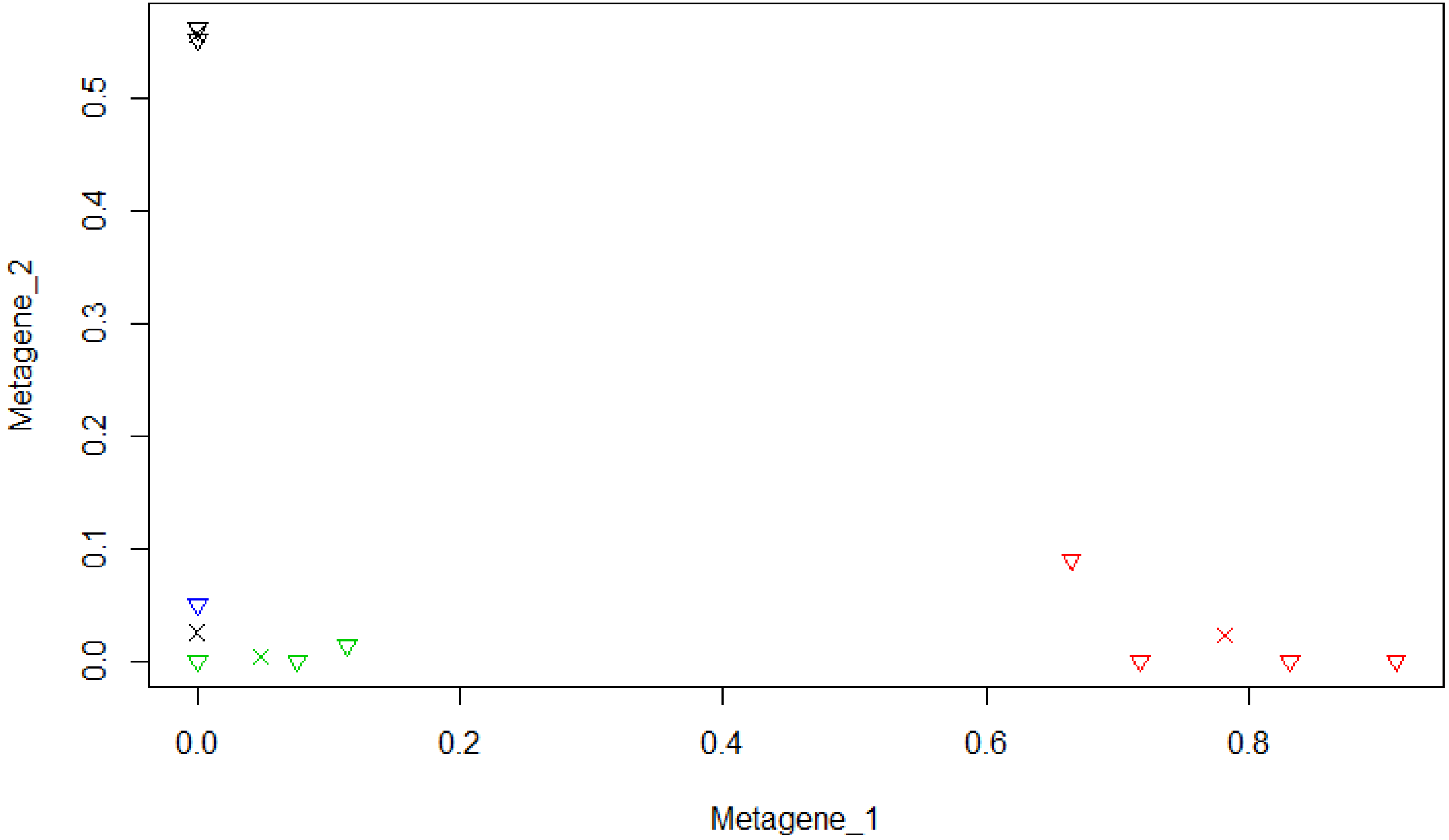
Different cluster labels obtained from several runs (m=10) of k-means clustering algorithms
Cluster labels (V_1, V_2, …, V_{10}) are **not unique**

Using plot3d() to visualise samples



Visualising the original samples when they coloured by the subgroups obtained from k-means clustering

After running k-means on these 12 samples



Any Questions?