

# **COMP47460 Tutorial**

## **Clustering**

**Aonghus Lawlor**  
**Derek Greene**

**School of Computer Science**  
**Autumn 2016**



# Tutorial Q1(a)

---

The dataset contains 10 examples represented by 4 numeric features.

These examples have been randomly assigned to two clusters in order to initialise the k-Means algorithm.

The assignments are as follows:

$$C1 = \{ x1, x3, x7, x8 \}$$

$$C2 = \{ x2, x4, x5, x6, x9, x10 \}$$

	f1	f2	f3	f4
x1	5.1	3.8	1.6	0.2
x2	4.6	3.2	1.4	0.2
x3	5.3	3.7	1.5	0.2
x4	5	3.3	1.4	0.2
x5	7	3.2	4.7	1.4
x6	6.4	3.2	4.5	1.5
x7	6.9	3.1	4.9	1.5
x8	5.5	2.3	4	1.3
x9	6.5	2.8	4.6	1.5
x10	5.7	2.8	4.5	1.3

Based on the data and cluster assignments, calculate the centroid vector for each cluster.

# Tutorial Q1(a)

- Recall -  $k$ -Means objective:

Centroid = mean of examples in cluster

$$SSE(\mathcal{C}) = \sum_{c=1}^k \sum_{x_i \in C_c} D(x_i, \mu_c)^2 \quad \text{where} \quad \mu_c = \frac{\sum_{x_i \in C_c} x_i}{|C_c|}$$

	f1	f2	f3	f4
x1	5.1	3.8	1.6	0.2
x2	4.6	3.2	1.4	0.2
x3	5.3	3.7	1.5	0.2
x4	5	3.3	1.4	0.2
x5	7	3.2	4.7	1.4
x6	6.4	3.2	4.5	1.5
x7	6.9	3.1	4.9	1.5
x8	5.5	2.3	4	1.3
x9	6.5	2.8	4.6	1.5
x10	5.7	2.8	4.5	1.3

Cluster 1	f1	f2	f3	f4
x1	5.1	3.8	1.6	0.2
x3	5.3	3.7	1.5	0.2
x7	6.9	3.1	4.9	1.5
x8	5.5	2.3	4	1.3
Centroid 1	5.70	3.23	3.00	0.80

Cluster 2	f1	f2	f3	f4
x2	4.6	3.2	1.4	0.2
x4	5	3.3	1.4	0.2
x5	7	3.2	4.7	1.4
x6	6.4	3.2	4.5	1.5
x9	6.5	2.8	4.6	1.5
x10	5.7	2.8	4.5	1.3
Centroid 2	5.87	3.08	3.52	1.02

$$C1 = \{ x1, x3, x7, x8 \}$$

$$C2 = \{ x2, x4, x5, x6, x9, x10 \}$$

(rounded to 2 decimal places)

# Tutorial Q1(b)

---

- Based on the centroids calculated above, which clusters will the examples  $x1$  and  $x10$  next be assigned to? Calculate distances using the Euclidean distance measure.

	f1	f2	f3	f4
<b>x1</b>	5.10	3.80	1.60	0.20
<b>Centroid 1</b>	5.70	3.23	3.00	0.80
<b>Centroid 2</b>	5.87	3.08	3.52	1.02

$$D(x, \mu) = \sqrt{\sum_{l=1}^m (x_l - \mu_l)^2}$$

$$D(x1, C1) \quad \sqrt{(5.10 - 5.70)^2 + (3.80 - 3.22)^2 + (1.60 - 3.00)^2 + (0.20 - 0.80)^2} = 1.74$$

$$D(x1, C2) \quad \sqrt{(5.10 - 5.87)^2 + (3.80 - 3.08)^2 + (1.60 - 3.52)^2 + (0.20 - 1.02)^2} = 2.33$$

$$D(x1, C1) = 1.74 \quad D(x1, C2) = 2.33 \quad \Rightarrow \quad \text{Assign to C1}$$

# Tutorial Q1(b)

---

- Based on the centroids calculated above, which clusters will the examples  $x1$  and  $x10$  next be assigned to? Calculate distances using the Euclidean distance measure.

	f1	f2	f3	f4
<b>x10</b>	5.70	2.80	4.50	1.30
<b>Centroid 1</b>	5.70	3.23	3.00	0.80
<b>Centroid 2</b>	5.87	3.08	3.52	1.02

$$D(x, \mu) = \sqrt{\sum_{l=1}^m (x_l - \mu_l)^2}$$

$$D(x10, C1) = \sqrt{(5.70 - 5.70)^2 + (2.80 - 3.22)^2 + (4.50 - 3.00)^2 + (1.30 - 0.80)^2} = 1.64$$

$$D(x10, C2) = \sqrt{(5.70 - 5.87)^2 + (2.80 - 3.08)^2 + (4.50 - 3.52)^2 + (1.30 - 1.02)^2} = 1.07$$

$$D(x10, C1) = 1.64 \quad D(x10, C2) = 1.07 \quad \Rightarrow \quad \text{Assign to C2}$$



# Tutorial Q2

---

- If the cluster  $C1 = \{x1, x3\}$ , use the Euclidean distance measure to calculate the distances between the example  $x2$  and cluster  $C1$  based on *single*, *complete*, and *average linkage*.

	f1	f2
x1	1.3	1.5
x2	0.5	2.4
x3	0.0	3.0

## Step 1: Calculate Euclidean distances

$$D(x1, x2) = 1.20$$

$$D(x1, x3) = 1.98$$

$$D(x2, x3) = 0.78$$

## Step 2: Calculate linkage metrics

$$\text{Single: } D(x2, C1) = \min(1.20, 0.78) = 0.78$$

$$\text{Complete: } D(x2, C1) = \max(1.20, 0.78) = 1.20$$

$$\text{Average: } D(x2, C1) = (1.20 + 0.78) / 2 = 0.99$$

# Tutorial Q3

---

- The following table depicts a pairwise distance matrix for 5 examples.
- Calculate the dendrogram representing the agglomerative hierarchical clustering of these examples based on the single-linkage method.
- The answer should illustrate the distance matrices originating from each clustering step.

e.g.  $D(x_3, x_1) = 6$   
and  $D(x_1, x_3) = 6$

	x1	x2	x3	x4	x5
x1	0				
x2	2	0			
x3	6	5	0		
x4	10	9	4	0	
x5	9	8	5	3	0

# Tutorial Q3

	x1	x2	x3	x4	x5
x1	0				
x2	2	0			
x3	6	5	0		
x4	10	9	4	0	
x5	9	8	5	3	0

**1** Start with everything in its own cluster:

Clusters:  $\{x1\}$ ,  $\{x2\}$ ,  $\{x3\}$ ,  $\{x4\}$ ,  $\{x5\}$

Identify nearest pair via single linkage

Min distance  $\Rightarrow D(x1, x2) = 2$

Merge:  $C1 = \{x1, x2\}$

**2** Clusters:  $C1$ ,  $\{x3\}$ ,  $\{x4\}$ ,  $\{x5\}$

Calculate distance matrix via single linkage

e.g.  $D(C1, x3) = \min(6, 5)$

Min distance  $\Rightarrow D(x4, x5) = 3$

Merge:  $C2 = \{x4, x5\}$

	C1	x3	x4	x5
C1	0			
x3	5	0		
x4	9	4	0	
x5	8	5	3	0

**3** Clusters:  $C1$ ,  $\{x3\}$ ,  $C2$

Calculate distance matrix via single linkage

e.g.  $D(C1, C2) = \min(10, 9, 9, 8) = 8$

Min distance  $\Rightarrow D(C2, x3) = 4$

Merge:  $C3 = \{x3, x4, x5\}$

	C1	x3	C2
C1	0		
x3	5	0	
C2	8	4	0



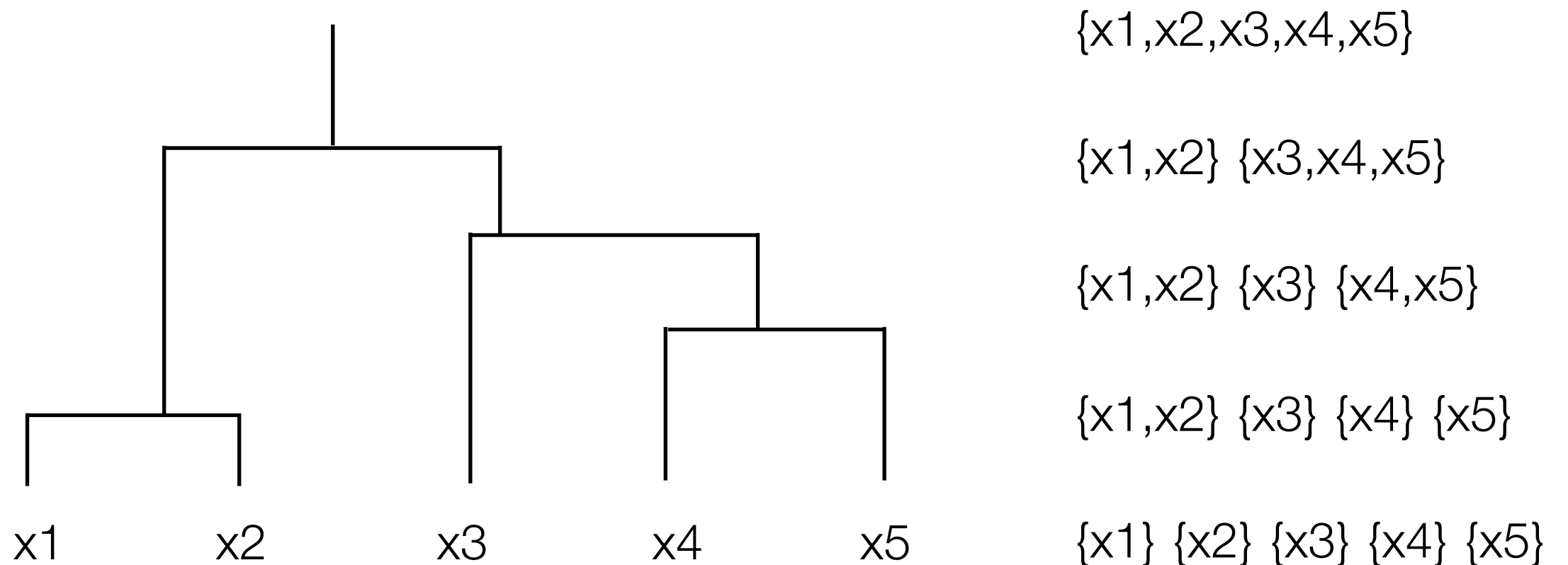
# Tutorial Q3

---

**4** Clusters:  $C1$ ,  $C3$  where  $C1 = \{x1, x2\}$ ,  $C3 = \{x3, x4, x5\}$

Only 2 clusters remain, so merge into root node  $C4$

Construct dendrogram based on the merges at each level...



# Tutorial Q4

---

In Weka, apply  $k$ -Means with Euclidean distance to the Iris ARFF dataset. Report the Within cluster sum of squared errors (SSE) for runs with different numbers of clusters:  $k=2$ ,  $k=3$  and  $k=4$ .

# Tutorial Q4

- In *Cluster* tab, choose *SimpleKMeans* as the clusterer. Change options for *numClusters* to 2, 3, 4.

The screenshot shows the Weka Explorer interface with the 'Cluster' tab selected. The 'Clusterer' dropdown is set to 'SimpleKMeans' with various options. The 'Cluster mode' section has 'Use training set' selected. The 'Clusterer output' pane shows the results of the clustering process, including the number of iterations, within-cluster sum of squared errors, initial starting points, final cluster centroids, and clustered instances.

**Clusterer**

Choose **SimpleKMeans** -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.Euclid"

**Cluster mode**

- ☒ Use training set
- ☐ Supplied test set Set...
- ☐ Percentage split % 66
- ☐ Classes to clusters evaluation (Nom) class
- ☒ Store clusters for visualization

Ignore attributes

Start Stop

**Result list (right-click for options)**

13:13:26 - SimpleKMeans

**Clusterer output**

kMeans  
=====

Number of iterations: 7  
Within cluster sum of squared errors: 62.1436882815797

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor  
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data (150.0)	Cluster# 0 (100.0)	1 (50.0)
sepalength	5.8433	6.262	5.006
sepalwidth	3.054	2.872	3.418
petallength	3.7587	4.906	1.464
petalwidth	1.1987	1.676	0.244
class	Iris-setosa Iris-versicolor		Iris-setosa

Time taken to build model (full training data) : 0.04 seconds

=== Model and evaluation on training set ===

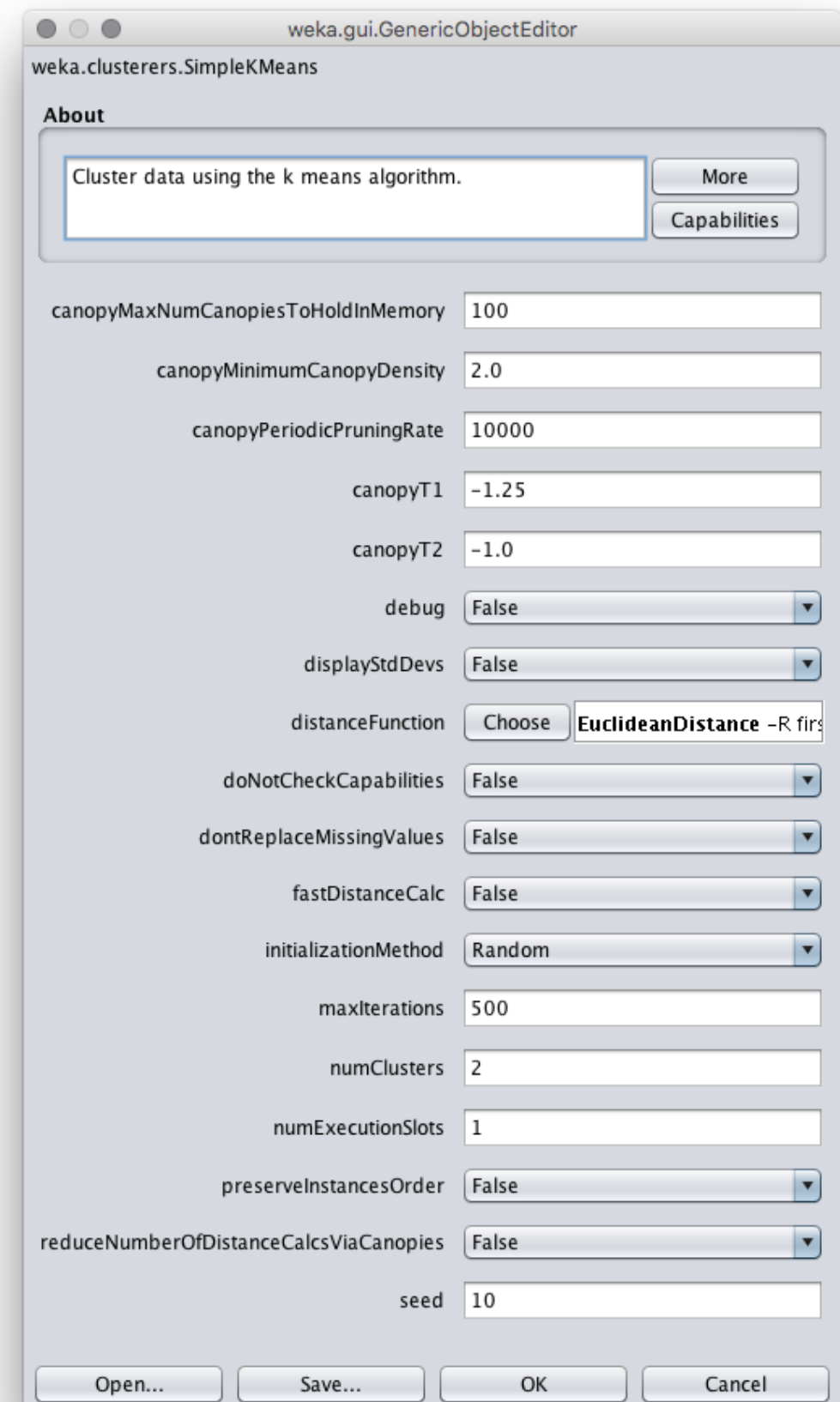
Clustered Instances

0	100 ( 67%)
1	50 ( 33%)

Status: OK Log x 0

# Tutorial Q4

- In the *Cluster* tab, choose *SimpleKMeans* as the clusterer.
- Change options for *numClusters* to 2, 3, 4.
- Use the default random *seed* (10).



# Tutorial Q4

---

- Report the within cluster *sum of squared errors* (SSE) for runs with different numbers of clusters:  $k=2$ ,  $k=3$  and  $k=4$ .
- Use default random seed ( $seed=10$ )

numClusters=2

Within cluster sum of squared errors: 62.1436882815797

numClusters=3

Within cluster sum of squared errors: 7.817456892309574

numClusters=4

Within cluster sum of squared errors: 6.613823274690356

# Tutorial Q4

---

- Repeat the above process again, but change the random seed parameter for k-Means. Are the SSE scores identical? If not, explain why not.

Set random seed to - e.g seed = 100

Changing random seed affects initial random clusters

numClusters=2

Within cluster sum of squared errors: 62.143688281579706

numClusters=3

Within cluster sum of squared errors: 60.90827498962252

numClusters=4

Within cluster sum of squared errors: 6.856549502288228