COMP47750 Tutorial

Clustering

Pádraig Cunningham

School of Computer Science

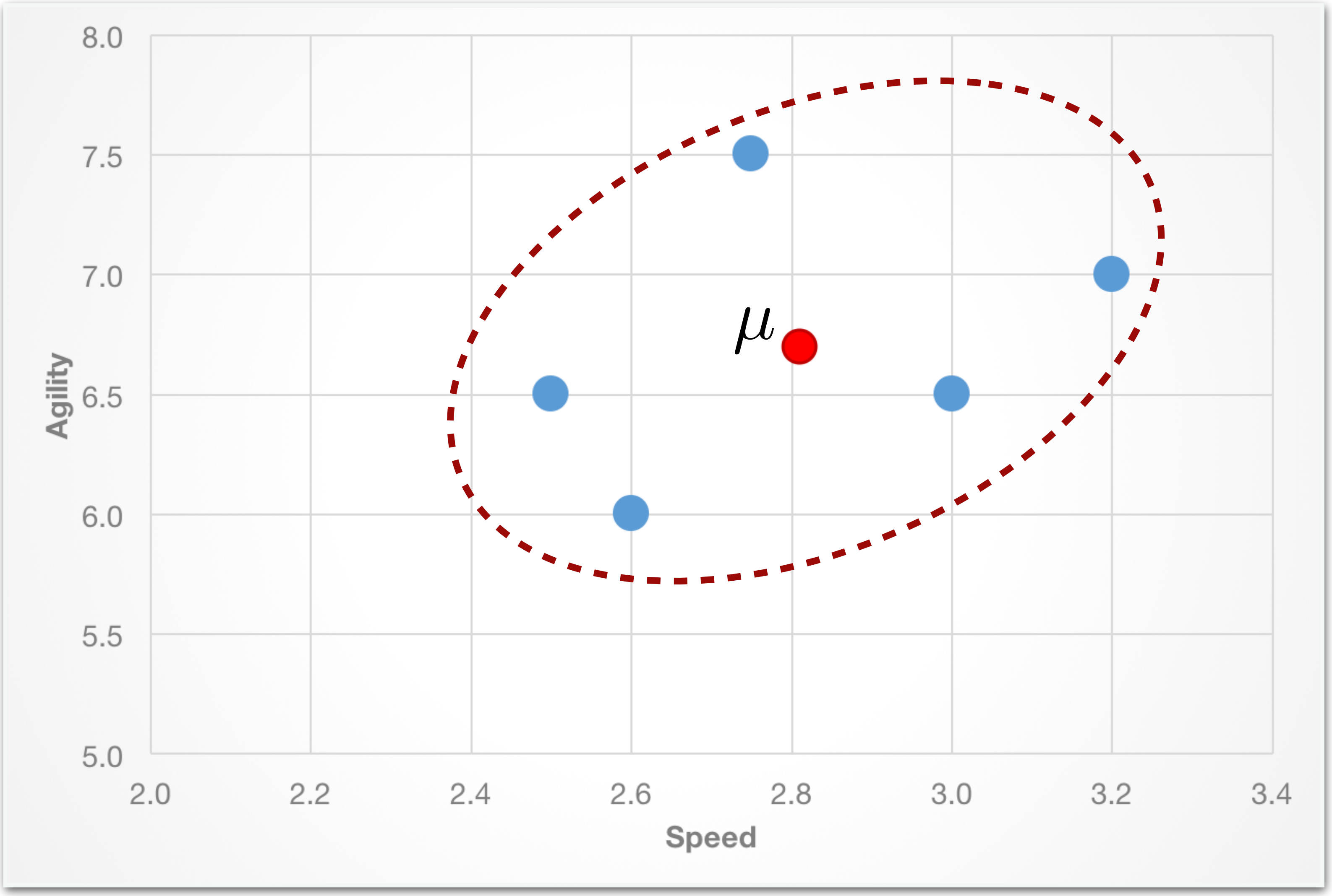
Reminder: *k*-Means Clustering

- Centroid: The mean vector of all items assigned to a given cluster (i.e. the mean of their feature vectors).

Athlete	*Speed*	*Agility*
1	2.6	6.0
2	3.0	6.5
3	2.5	6.5
4	3.2	7.0
5	2.8	7.5
Centroid	2.82	6.7

$(2.6 + 3.0 + 2.5 + 3.2 + 2.8)/5$
$= 2.82$

$(6.0 + 6.5 + 6.5 + 7.0 + 7.5)/5$
$= 6.7$



Reminder: *k*-Means Clustering

- **Goal:** Minimise distances between the items and their nearest centroid - i.e. minimisation of *sum-of-squared error* (SSE):

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

- In the standard algorithm, *D* is measured using Euclidean distance:

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

sum of squared difference over all *m* feature values

- *k*-Means tries to reduce SSE via a two step iterative process:
  1) Reassign items to their nearest cluster centroid
  2) Update the centroids based on the new assignments
- Repeatedly apply these two steps until the algorithm converges to a final result.

Reminder: Agglomerative Clustering

Algorithm Inputs:

- *Distance matrix* $\mathbf{D}$, specifying the distance between each pair of items in the data, computed using some appropriate measure (e.g. Euclidean).
- *Cluster metric* which helps decide which pair of clusters to merge at each step, using values from $\mathbf{D}$.

Algorithm summary:

1. Assign every item to its own cluster, each just containing that item. These are the “leaf nodes” of the tree.
2. Find the closest (i.e. most similar) pair of clusters, according to the cluster metric, and merge them into a single cluster, so that now you have one less cluster.
3. Compute distances (similarities) between the new cluster and each of the remaining old clusters.
4. Repeat from Step 2 until all items are clustered into a single cluster. This is the “root node” of the tree.

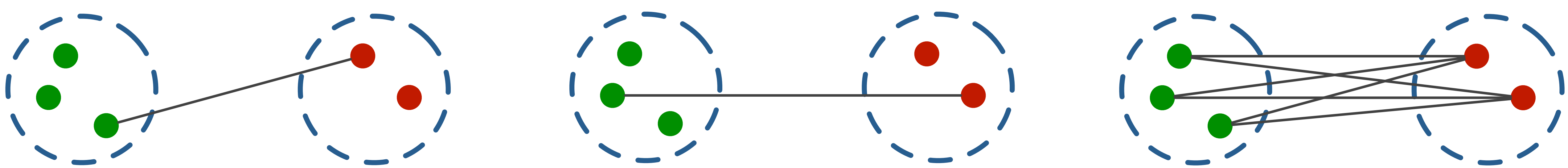
Reminder: Agglomerative Clustering

- Cluster metrics specify how we use values from $\mathbf{D}$ to measure the distance between two clusters.
- Formulae for cluster distance metrics, where D_{ij} is the distance between the *i*-th and *j*-th items in distance matrix $\mathbf{D}$:

Single linkage
$$d(C_a, C_b) = \min_{x_i \in C_a, x_j \in C_b} D_{ij}$$

Complete linkage
$$d(C_a, C_b) = \max_{x_i \in C_a, x_j \in C_b} D_{ij}$$

Average linkage
$$d(C_a, C_b) = \frac{\sum_{x_i \in C_a} \sum_{x_j \in C_b} D_{ij}}{|C_a|\,|C_b|}$$



Single Linkage   Complete Linkage   Average Linkage

Tutorial Q1(a)

The dataset contains 10 examples described 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)

The dataset contains 10 examples described 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

C1 = { x1, x3, x7, x8 }

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

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

(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
x1	5.10	3.80	1.60	0.20
Centroid 1	5.70	3.23	3.00	0.80
Centroid 2	5.87	3.08	3.52	1.02

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

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

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

D(x1,C1) = 1.74 *D(x1,C2) = 2.33* => 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
x10	5.70	2.80	4.50	1.30
Centroid 1	5.70	3.23	3.00	0.80
Centroid 2	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 *D(x10,C2) = 1.07* => Assign to C2

Tutorial Q1(c)

- Why is it not possible to use *k*-means clustering with categorical data?

If the features are categorical (rather than numeric) then it is not possible to calculate means which is an essential step in determining the cluster centroids.

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

Single: D(x2,C1) = min(1.20,0.78) = 0.78

Complete: D(x2,C1) = max(1.20,0.78) = 1.20

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(x3,x1) = 6
and D(x1,x3) = 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 ⇒ 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 ⇒ 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 ⇒ D(C2,x3) = 4

Merge: C3 = {x3,x4,x5}

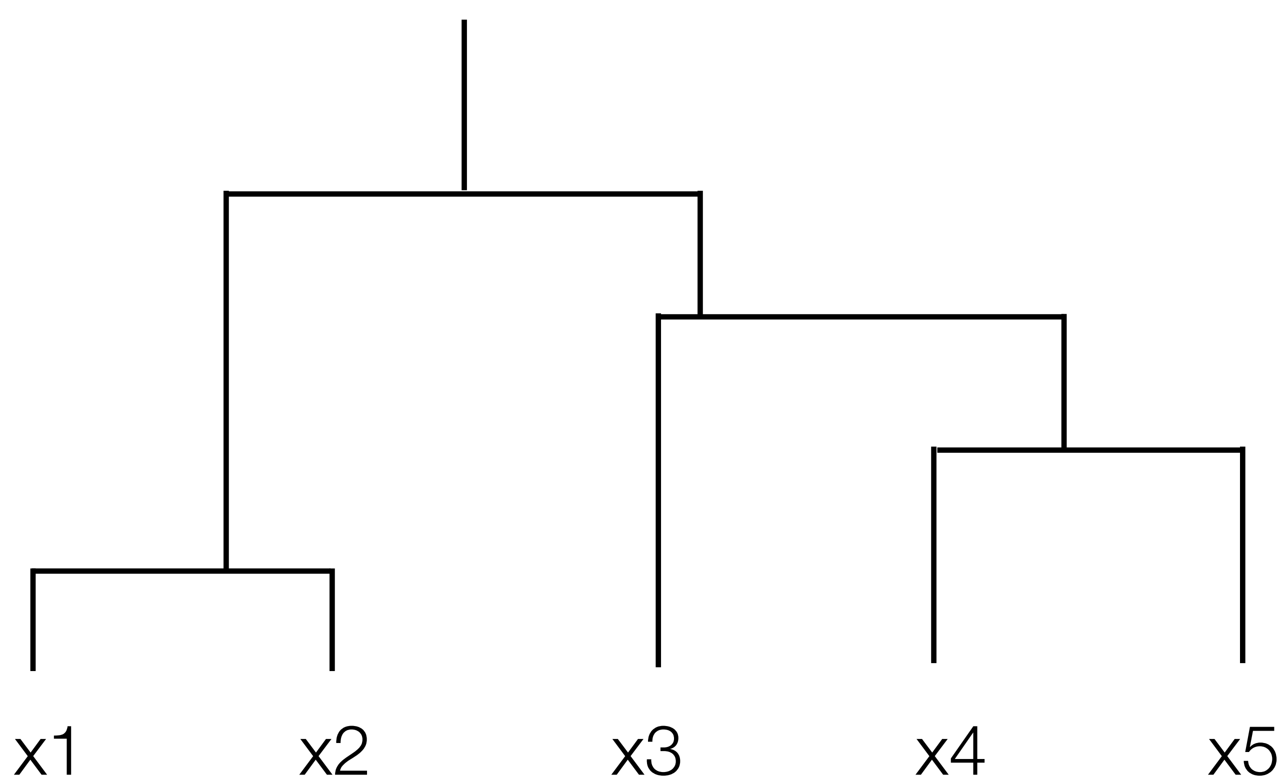
	C1	x3	C2
C1	0		
x3	5	0	
C2	8	4	0

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…



{x1,x2,x3,x4,x5}

{x1,x2} {x3,x4,x5}

{x1,x2} {x3} {x4,x5}

{x1,x2} {x3} {x4} {x5}

{x1} {x2} {x3} {x4} {x5}