Unsupervised Learning: Clustering Using Excel

Rao Vemuri

Clustering

- Clustering is a technique for finding similarity groups in data, called **clusters**. i.e.,
  - Group data that are similar to (near) each other in one cluster
  - Group data that are very different (far away) from each other into a different cluster.
- Clustering is often called an **unsupervised learning** as no class labels denoting grouping are given

An illustration

- The data set has three natural groups of data points, i.e., 3 natural clusters.

What is Clustering For?

- Let us see some real-life examples
- Example 1: Group people of similar sizes together to manufacture “small”, “medium” and “large” T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.
- Example 2: In marketing, group customers according to their similarities
  - To do targeted marketing.

What is Clustering for? (cont…)

- Example 3: Given a collection of documents, organize them according to their similarity
  - In recent years, due to the rapid increase of online documents, text clustering became important.
- In fact, clustering is one of the most utilized data mining techniques
  - It has a long history, and used in many fields
    - e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.

Aspects of Clustering

- A clustering algorithm
- A distance (similarity or dissimilarity) metric
- Clustering quality
  - Inter-clusters distance $\Rightarrow$ maximized
  - Intra-clusters distance $\Rightarrow$ minimized
- The quality of a clustering depends on the algorithm, the distance metric, and the application.

K-means clustering

- 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
- Let the 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.

K-means algorithm

- Given *k*, the *k-means* algorithm works as follows:
  1) Randomly choose *k* data points (seeds) to be the initial centroids (cluster centers)
  2) Assign each data point to its closest centroid
  3) Re-compute the centroids using the current cluster memberships.
  4) If a convergence criterion is not met, go to 2.

K-means algorithm – (cont …)

Algorithm *k*-means(*k*, *D*)

```
1  Choose k data points as the initial centroids (cluster centers)
2  repeat
3      for each data point x ∈ D do
4          compute the distance from x to each centroid;
5          assign x to the closest centroid          // a centroid represents a cluster
6      endfor
7      re-compute the centroids using the current cluster memberships
8  until the stopping criterion is met
```

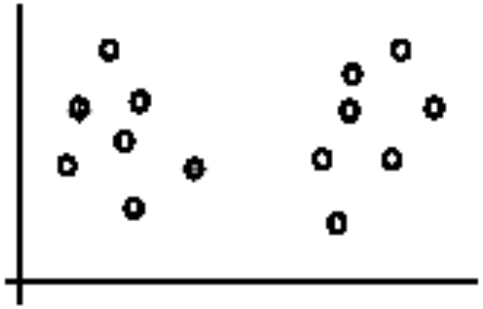
Stopping/convergence Criteria

1. No (or minimum) re-assignments of data points to different clusters, OR
2. No (or minimum) change of centroids, OR
3. Minimum decrease in the **sum of squared error** (SSE),

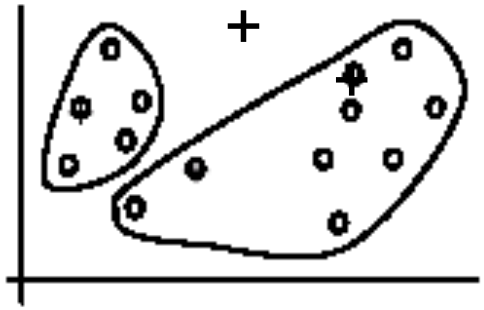
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2 \qquad (1)$$

- C_i is the jth cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j), and $dist(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

An Example with k =2



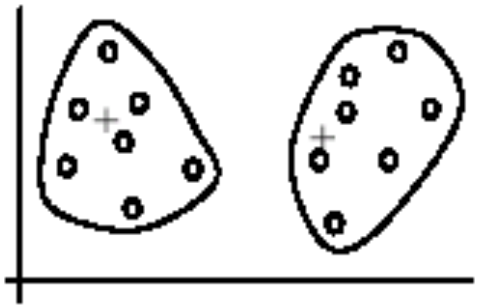
(A). Random selection of k centers



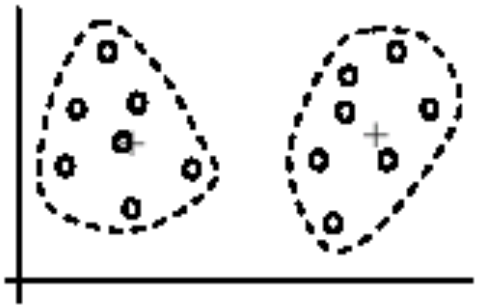
Iteration 1: (B). Cluster assignment

(C). Re-compute centroids

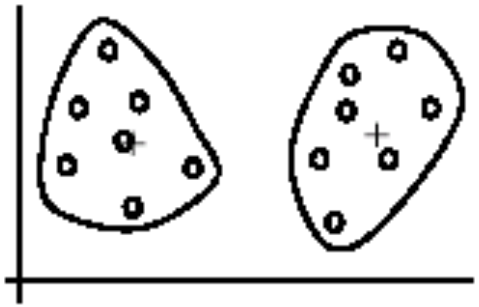
An example (cont …)



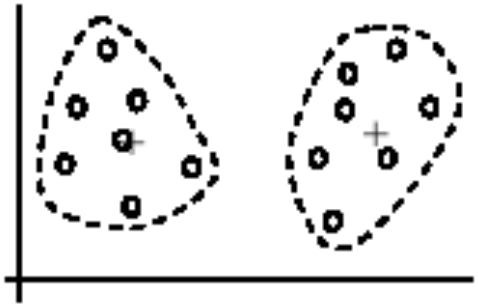
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

Excel Implementation

- Step 1: Start with a Dataset
- We have data on 15 individuals:
- Three attributes: x, y, z
- x = height, y = weight, z = wealth
- Notice we used scaled data

Data Set of 15 Cases, 3 Clusters, 3 Attributes

	Case	X	Y	Z
	1	4.40	4.57	2.29
	2	3.25	3.92	2.17
	3	3.10	4.25	2.40
	4	4.83	4.31	2.16
	5	3.63	3.60	1.67
Start1	6	3.26	1.64	1.48
	7	4.89	1.33	1.04
	8	4.50	2.01	1.28
Start2	9	4.99	2.47	2.60
	10	4.12	2.12	1.70
	11	2.21	2.51	4.17
	12	2.97	4.10	3.92
	13	2.40	2.45	4.46
	14	2.10	3.30	4.90
Start3	15	1.13	2.05	3.28
	Min	1.13	1.33	1.04
	Max	4.99	4.57	4.90
	Median	3.26	2.51	2.29

Start with Random Initial Cluster Centers

- Let us use k-means clustering method
- Let us assume that we wish to create 3 clusters (you can choose to cluster any number of clusters)
- For k-means clustering you typically pick some random starting points or seeds to get the analysis started.
- For these start points I have selected cases 6, 9 and 15: highlighted
- (These are the min, max and med of attribute x)
- – but any 3 random points would work.

Fig 4: Distance of each Data Point with Each Cluster Center

Case	Start1	Start2	Start3	Min	Initial Choice
1	10.54	4.84	18.07	4.84	2
2	5.64	5.30	9.19	5.30	2
3	7.65	6.75	9.50	6.75	2
4	10.01	3.58	20.05	3.58	2
5	4.00	3.97	11.25	3.97	2
6	-	4.91	7.95	-	1
7	2.94	3.75	19.67	2.94	1
8	1.72	2.18	15.40	1.72	1
9	4.91	-	15.54	-	2
10	1.02	1.67	11.47	1.02	1
11	9.09	10.19	2.18	2.18	3
12	12.10	8.50	8.02	8.02	3
13	10.28	10.19	3.17	3.17	3
14	15.79	14.31	5.15	5.15	3
15	7.95	15.54	-	-	3
			SSE	48.64	

Explanation of Table

- Referring to the table output – this is our first calculation in Excel and it generates our “initial choice” of clusters.
- Start 1 is the data for case 6
- start 2 is case 9
- start 3 is case 15.
- Note that the intersection of each of these gives a 0 (-) in the table.

Fig 5: How did we get 10.54 for Case 1, Start 1

	Case	X	Y	Z	
	1	4.40	4.57	2.29	
Start1	6	3.26	1.64	1.48	
	Difference	1.14	2.93	0.81	
	Sqaured	**1.31**	**8.58**	**0.66**	**10.54**

Remember that we have arbitrarily designated Case 6 to be our random start point for Cluster 1. We want to calculate the distance and we use the sum of squares method – as shown here. We calculate the difference between each of the three data points in the set, and then square the differences, and then sum them.
We can do it "mechanically" as shown here – but Excel has a built-in formula to use: SUMXMY2 – this is far more efficient to use.

Referring back to Figure 4, we then find the minimum distance for each case from each of the three start points – this tells us which cluster (1, 2 or 3) that the case is closest to – which is shown in the 'initial choice column'.

Step 4: Fig 6: Calculate the Mean of Each Cluster

- We have now allocated each case to its initial cluster – and we can lay that out using an IF statement in the table. At the bottom of the table, we have the mean (average) of each of these cases.).

		Cluster 1			Cluster 2			Cluster 3		
Current	Case	X	Y	Z	X	Y	Z	X	Y	Z
2	1				4.40	4.57	2.29			
2	2				3.25	3.92	2.17			
2	3				3.10	4.25	2.40			
2	4				4.83	4.31	2.16			
2	5				3.63	3.60	1.67			
1	6	3.26	1.64	1.48						
1	7	4.89	1.33	1.04						
1	8	4.50	2.01	1.28						
2	9				4.99	2.47	2.60			
1	10	4.12	2.12	1.70						
3	11							2.21	2.51	4.17
3	12							2.97	4.10	3.92
3	13							2.40	2.45	4.46
3	14							2.10	3.30	4.50
3	15							1.13	2.05	3.28
	Mean	4.19	1.77	1.37	4.03	3.85	2.21	2.16	2.88	4.15

Step 5: Repeat Step 3 – the Distance from the revised mean

- The cluster analysis process now becomes a matter of repeating Steps 4 and 5 (iterations) until the clusters stabilize.
- Each time we use the revised mean for each cluster. Therefore, Figure 7 shows our second iteration – but this time we are using the means generated at the bottom of Figure 6 (instead of the start points from Figure 1).

Fig 7

Case	Cluster 1	Cluster 2	Cluster 3	Min	Revised Choice
1	8.71	0.66	11.33	0.66	2
2	6.12	0.63	6.14	0.63	2
3	8.35	1.06	5.82	1.06	2
4	7.42	0.84	13.11	0.84	2
5	3.73	0.53	8.82	0.53	2
6	0.90	6.01	9.85	0.90	1
7	0.79	8.49	19.52	0.79	1
8	0.16	4.50	14.48	0.16	1
9	2.61	2.96	10.56	2.61	1
10	0.23	3.26	10.40	0.23	1
11	12.29	8.96	0.14	0.14	3
12	13.42	4.12	2.19	2.19	3
13	13.21	9.70	0.34	0.34	3
14	19.14	11.26	0.75	0.75	3
15	13.10	12.82	2.52	2.52	3
			SSE	14.35	

Study this at Home

- You can now see that there has been a slight change in cluster application, with case 9 – one of our starting points – being reallocated.
- You can also see sum of squared error (SSE) calculated at the bottom – which is the sum of each of the minimum distances.
- Our goal is to now repeat Steps 4 and 5 until the SSE only shows minimal improvement and/or the cluster allocation changes are minor on each iteration.

Homework

- Repeat this problem and reproduce these results
- Use either Excel, Google Sheets or Python program
- Using a calculator is a last resort; it is too tedious