Unsupervised Learning and K-Means Clustering

Data Science Dojo

Unsupervised Learning

- Trying to find hidden structure in unlabelled data
- No label
- No error or reward signal to evaluate a potential solution

K-Means Clustering Algorithm

Suppose set of data points: $\{ x_1, x_2, x_3 \ldots\ldots\ldots x_n\}$

- Step 1: Place centroids at random locations
  - $c_1, c_2, \ldots. c_k$
- Step 2: Repeat until convergence:

{ for each point x_i $\rightarrow$ find nearest centroid c_j (eg. Using Euclidean distance) $\rightarrow$ assign the point x_i to cluster j

for each cluster $j = 1\ldots k$ $\rightarrow$ calculate new centroid c_j

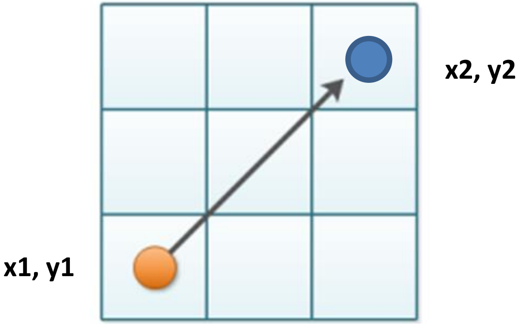
c_j=mean of all points x_i assigned to cluster j in previous step

}

- Step 3: Stop when none of the cluster assignments change

Euclidean Distance

Determine intra- and inter-cluster similarity



$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Minimise the sum of the Euclidean distances for each cluster

number of clusters

number of cases

centroid for cluster *j*

case *i*

objective function ← $$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \underbrace{\left\| x_i^{(j)} - c_j \right\|^2}_{\text{Distance function}}$$

datasciencedojo
data science for everyone

Choose number of clusters

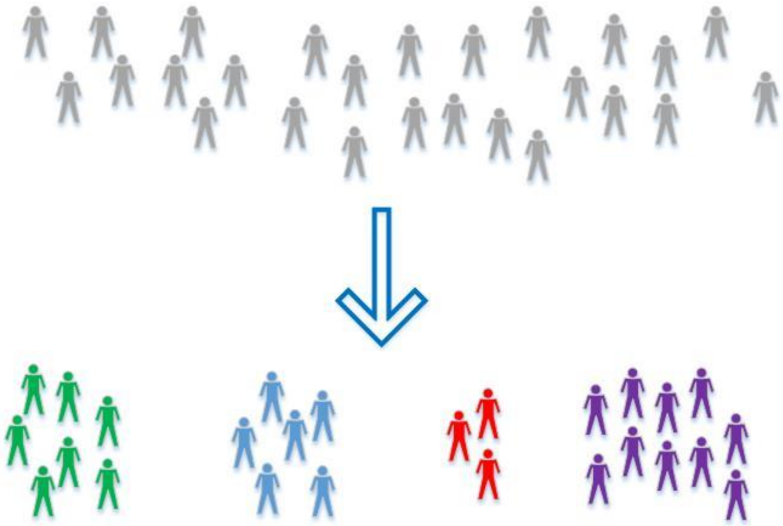
Example 1 (domain knowledge / practicalities): Clothing sizes

- Tailor-made for each person is too expensive
- One-size-fits-all: does not work!
- Groups people of similar sizes together to make "small", "medium", and "large" t-shirts

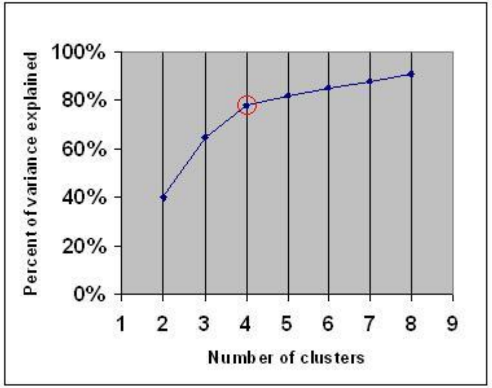
Choose number of clusters

Example 2 (via evaluation): Target marketing

- Subdivide market into distinct subsets of customers
- where any subset may conceivably be selected as a segment to be reached with a particular offer

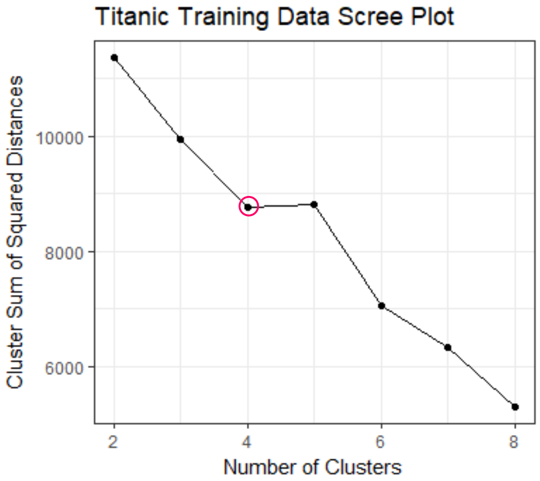


Finding K with Elbow Method



Option 1 - Percentage of variance explained as a function of the number of clusters.

Option 2 -Total of the squared distances of cluster point to center.



Goal - Choose a number of clusters so that adding another cluster doesn't give much better modelling of the data.

K-Means Clustering Algorithm

Suppose set of data points: $\{ x_1, x_2, x_3 \ldots\ldots\ldots x_n \}$

- Step 0: Decide the number of clusters, K=1,2,…k.
- Step 1: Place centroids at random locations
  - $c_1, c_2, \ldots. c_k$
- Step 2: Repeat until convergence:

{ for each point x_i → find nearest centroid c_j (eg. Euclidean distance)

→ assign the point x_i to cluster j

for each cluster j = 1…k → calculate new centroid c_j

c_j=mean of all points x_i assigned to cluster j in previous step

}

- Step 3: Stop when none of the cluster assignments change

Preparation

- Transform categorical variables into numeric
- Standardise
- Reduce dimensionality

Often called
“dummy variables” or
“one-hot encoding”

Age	Pclass.1	Pclass.2	Pclass.3	Sex.female	Sex.male
19	0	1	0	0	1
28	1	0	0	1	0
64	0	0	1	0	1

K-Means Clustering Algorithm

Suppose set of data points: $\{ x_1, x_2, x_3 ....x_n \}$

- Step -1: Convert to numeric, standardise and reduce dimensionality
- Step 0: Decide the number of clusters, K=1,2,...k.
- Step 1: Place centroids at random locations
  - ➢ $c_1, c_2, ... c_k$
- Step 2: Repeat until convergence:

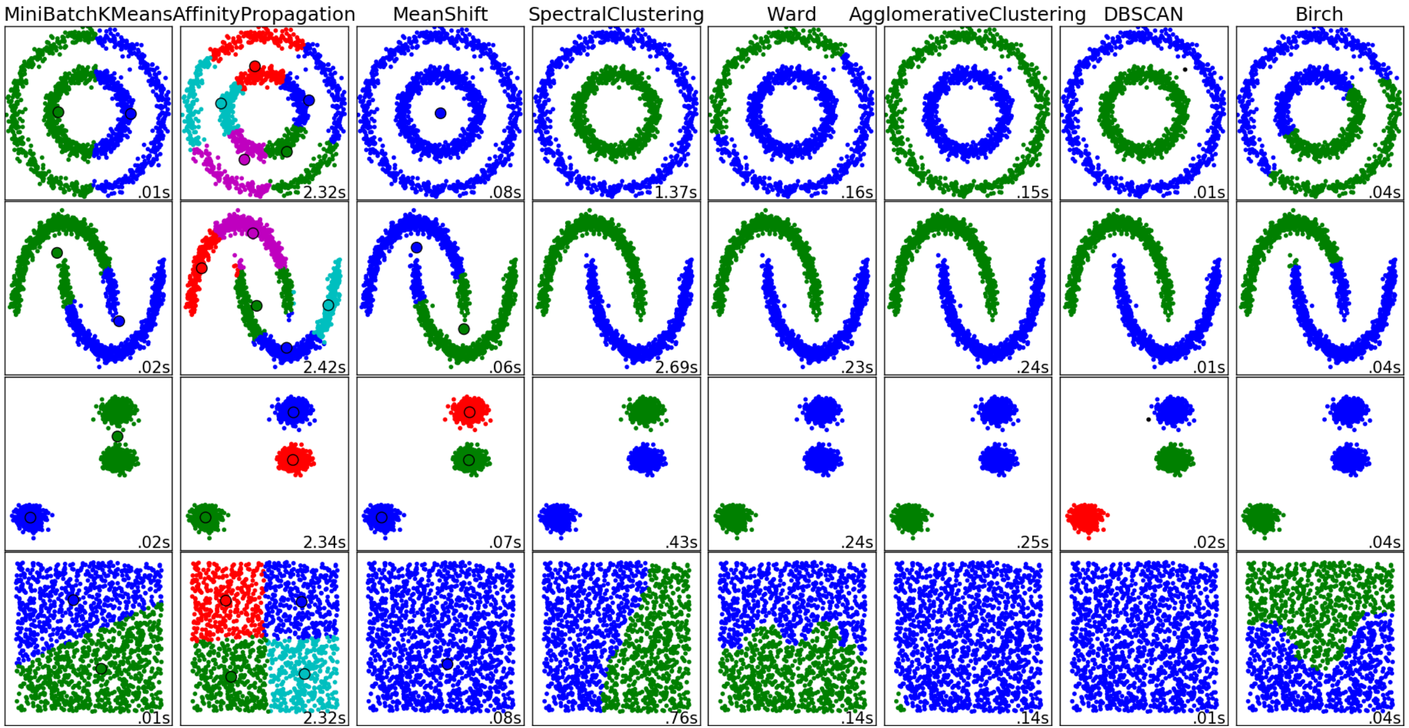
{ for each point x_i $\rightarrow$ find nearest centroid c_j (eg. Euclidean distance)

assign the point x_i to cluster j for each cluster j = 1...k $\rightarrow$ calculate new centroid c_j=mean of all points x_i assigned to cluster j in previous step $\rightarrow$

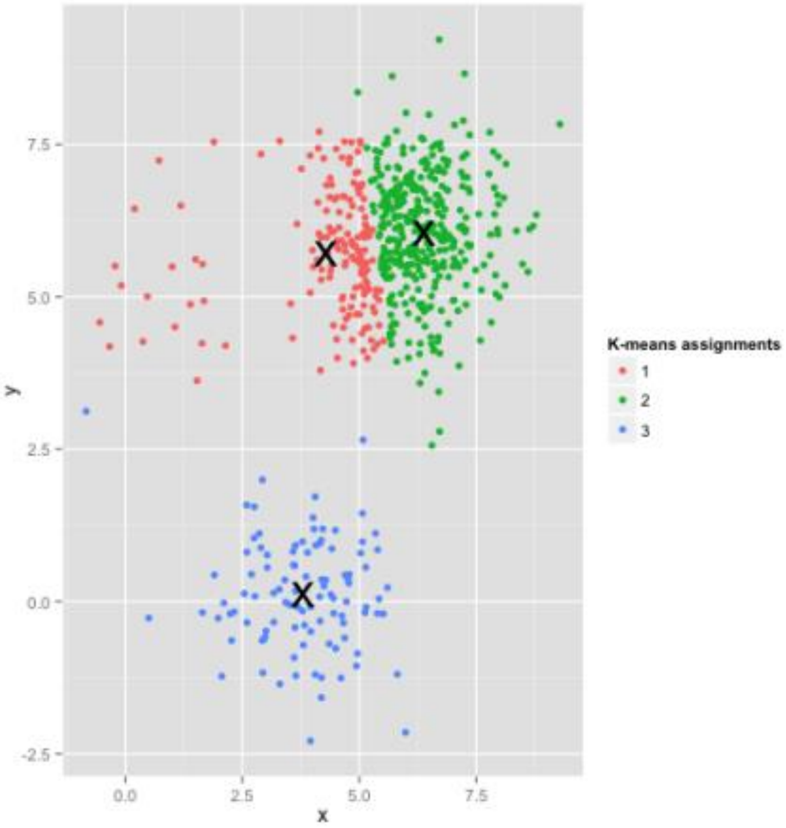
}

- Step 3: Stop when none of the cluster assignments change

Doesn’t do well on non-blobs



Doesn't do well when clusters are unevenly sized



data science for everyone

Strengths and Assumptions

- Strengths
  - Simple: easy to understand and to implement
  - Efficient: linear time, minimal storage
- Assumptions
  - Distribution of each variable is blob-like (remove outliers; try other clustering methods)
  - All variables have the same variance (standardise)
  - Equal prior probability of each cluster i.e. similar size (try different *k*s)

QUESTIONS