GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
CLUSTER ANALYSIS

DATA SCIENCE IN THE NEWS

I. CLUSTER ANALYSIS

	continuous	categorical
supervised	???	???
unsupervised	???	???

	continuous	categorical
supervised	regression	classification
unsupervised	dimension reduction	clustering

Q: What is a cluster?

Q: What is a cluster?

A: A group of similar data points.

Q: What is a cluster?

A: A group of similar data points.

The concept of similarity is central to the definition of a cluster, and therefore to cluster analysis.

Q: What is a cluster?

A: A group of similar data points.

The concept of similarity is central to the definition of a cluster, and therefore to cluster analysis.

In general, greater similarity between points leads to better clustering.

Q: What is the purpose of cluster analysis?

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing the data into groups.

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing the data into groups.

Clustering provides a *layer of abstraction* from individual data points.

Q: What is the purpose of cluster analysis?

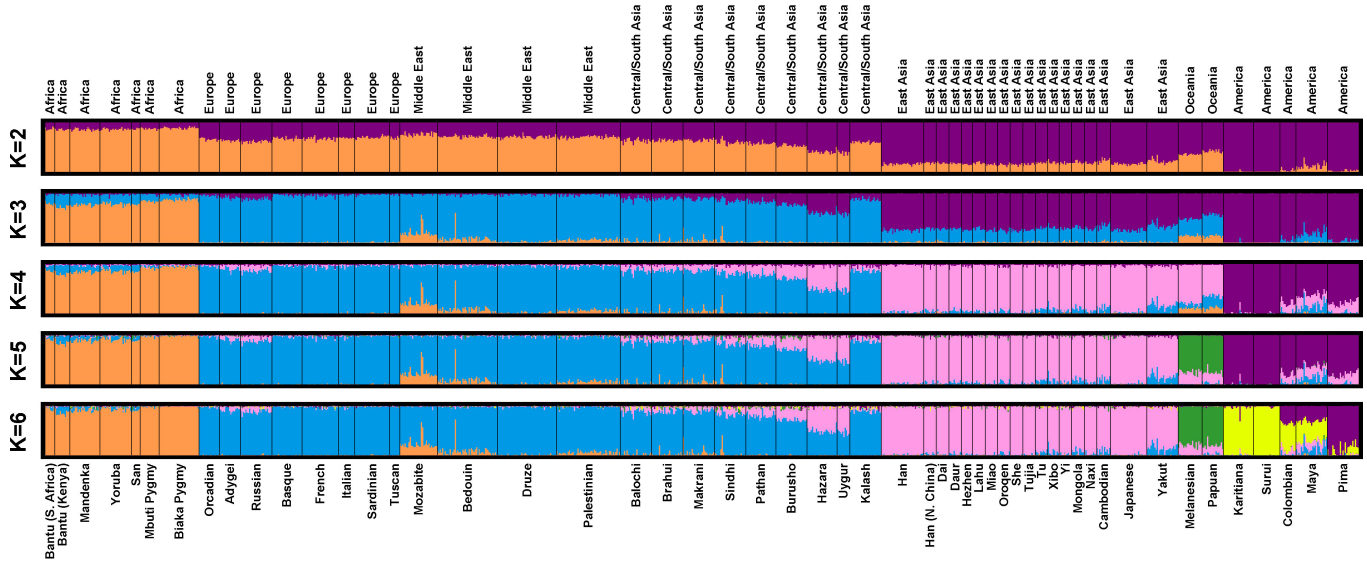
A: To enhance our understanding of a dataset by dividing the data into groups.

Clustering provides a *layer of abstraction* from individual data points.

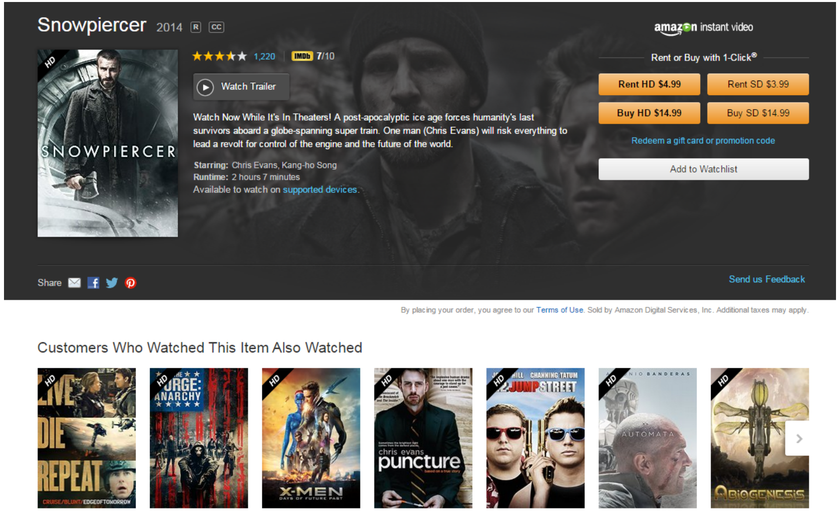
The goal is to extract and enhance the natural structure of the data

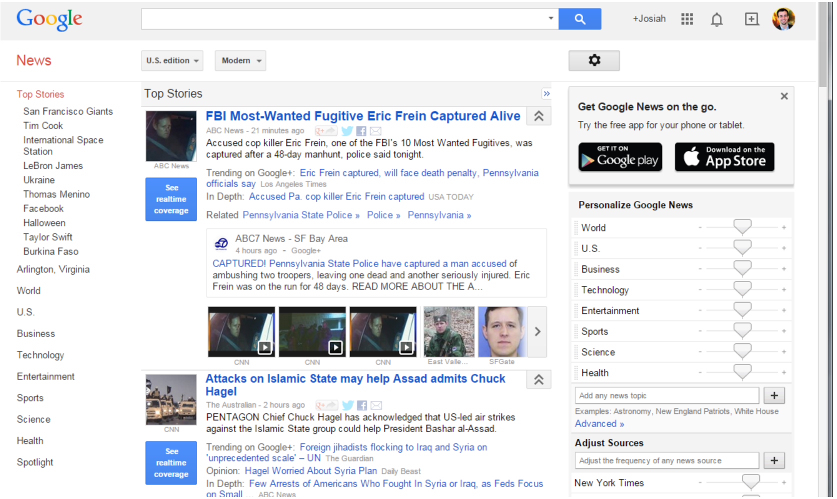
Clustering can be useful in a wide variety of domains, including genetics, consumer internet and business.

Clustering can be useful in a wide variety of domains, including **genetics**, consumer internet and business.



Clustering can be useful in a wide variety of domains, including genetics, **consumer internet** and business.





Clustering can be useful in a wide variety of domains, including genetics, consumer internet and **business**.

http://i.huffpost.com/gen/1563531/thumbs/o-GROCERY-STORE-facebook.jpg

There are many kinds of classification procedures. For our class, we will be focusing on K-means clustering, which is one of the most popular clustering algorithms.

K-means is an iterative method that partitions a data set into k clusters.

II. K-MEANS CLUSTERING

Q: How does the algorithm work?

1) choose k initial centroids (note that k is an input)

2) for each point:
   - find distance to each centroid
   - assign point to nearest centroid

3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

Q: How do you choose the initial centroid positions?

Q: How do you choose the initial centroid positions?

A: There are several options:

Q: How do you choose the initial centroid positions?

A: There are several options:

- randomly (but may yield divergent behavior)

Q: How do you choose the initial centroid positions?

A: There are several options:
- randomly (but may yield divergent behavior)
- perform alternative clustering task, use resulting centroids as initial k-means centroids

Q: How do you choose the initial centroid positions?

A: There are several options:

- randomly (but may yield divergent behavior)
- perform alternative clustering task, use resulting centroids as initial k-means centroids
- start with global centroid, choose point at max distance, repeat (but might select outlier)

Q: How do you determine which centroid a given point is most similar to?

Q: How do you determine which centroid a given point is most similar to?

The similarity criterion is determined by the measure we choose.

Q: How do you determine which centroid a given point is most similar to?

The similarity criterion is determined by the measure we choose.

In the case of k-means clustering, the similarity metric is the **Euclidian distance**:

Q: How do you determine which centroid a given point is most similar to?

The similarity criterion is determined by the measure we choose.

In the case of k-means clustering, the similarity metric is the **Euclidian distance**:

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{N} (x_{1i} - x_{2i})^2}$$

Q: How do we re-compute the positions of the centers at each iteration of the algorithm?

A: By calculating the centroid (i.e., the geometric center)

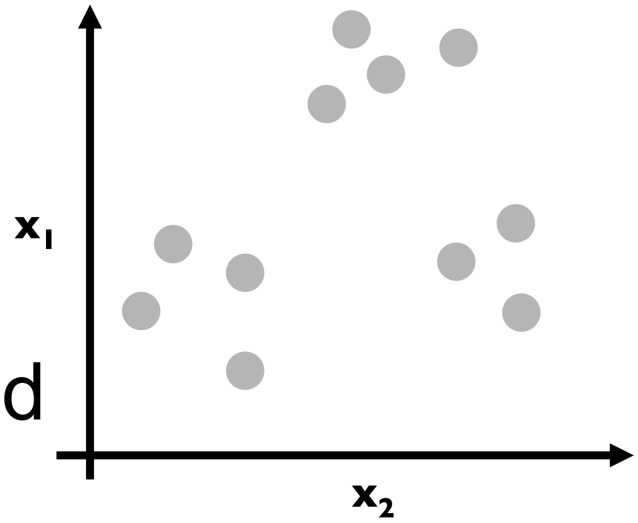
We iterate until some stopping criteria are met; in general, suitable convergence is achieved in a small number of steps.

We iterate until some stopping criteria are met; in general, suitable convergence is achieved in a small number of steps.

Stopping criteria can be based on the centroids (eg, if positions change by no more than ε) or on the points (eg, if no more than $x\%$ change clusters between iterations).

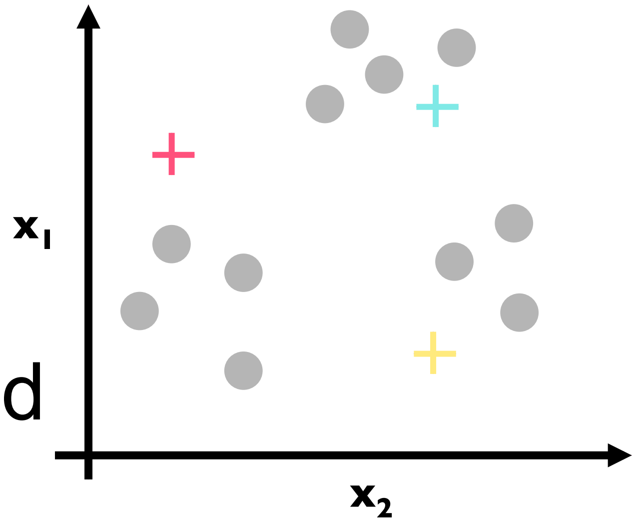
1) choose k initial centroids (note that k is an input)

2) for each point:
   - find distance to each centroid
   - assign point to nearest centroid



3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

1) choose k initial centroids (note that k is an input)

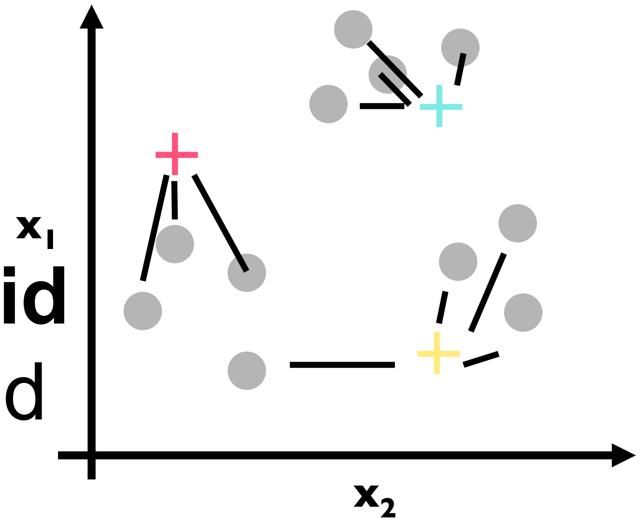


2) for each point:
   - find distance to each centroid
   - assign point to nearest centroid

3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

1) choose k initial centroids (note that k is an input)

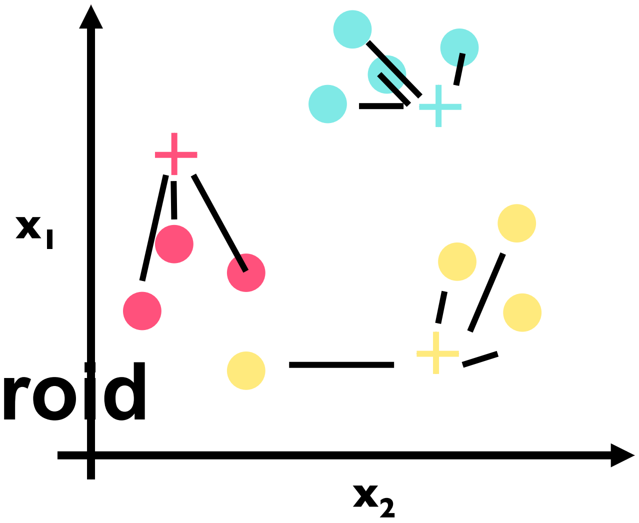
2) for each point:
- find distance to each centroid
- assign point to nearest centroid



3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

1) choose k initial centroids (note that k is an input)

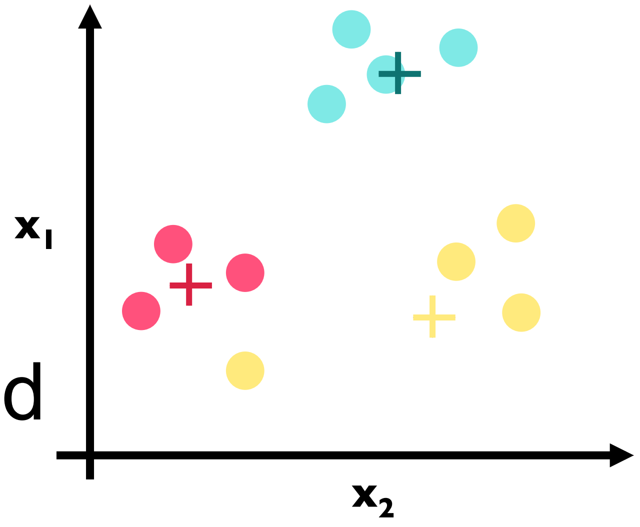
2) for each point:
- find distance to each centroid
- **assign point to nearest centroid**



3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

1) choose k initial centroids (note that k is an input)

2) for each point:
 - find distance to each centroid
 - assign point to nearest centroid

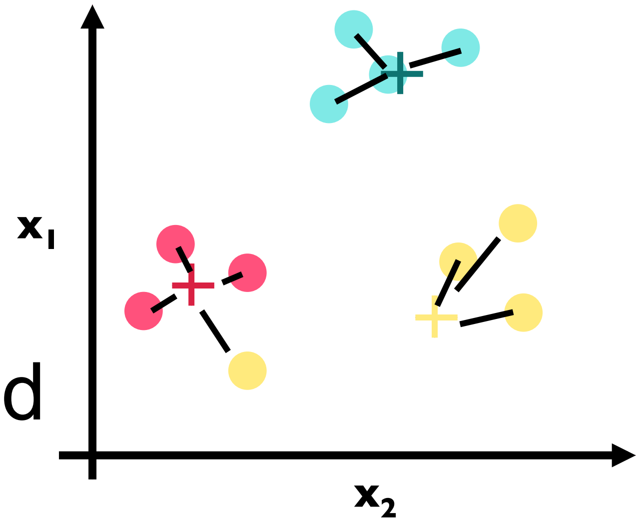


3) recalculate centroid positions

4) repeat steps 2-3 until stopping criteria met

1) choose k initial centroids (note that k is an input)

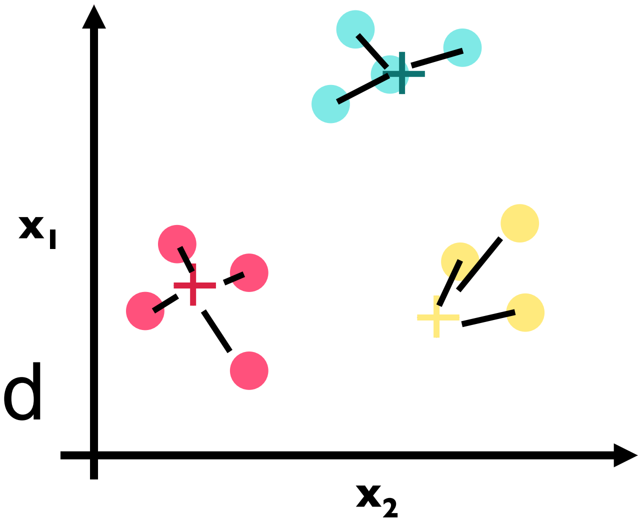
2) for each point:
   - find distance to each centroid
   - assign point to nearest centroid



3) recalculate centroid positions

4) repeat steps 2-3 until stopping criteria met

1) choose k initial centroids (note that k is an input)



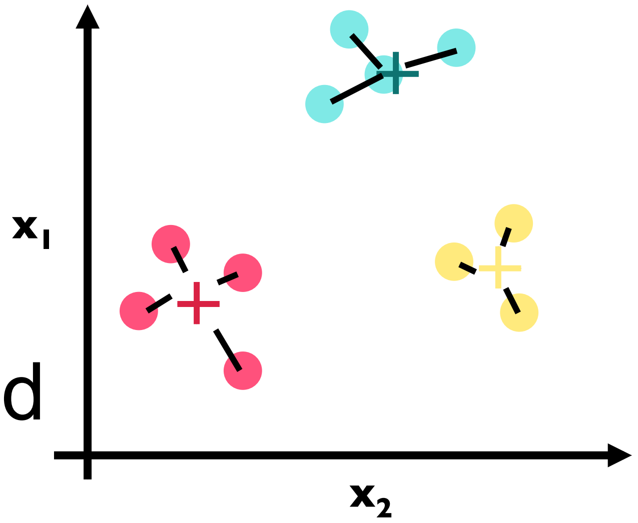
2) for each point:
- find distance to each centroid
- assign point to nearest centroid

3) recalculate centroid positions

4) repeat steps 2-3 until stopping criteria met

1) choose k initial centroids (note that k is an input)

2) for each point:
  - find distance to each centroid
  - assign point to nearest centroid

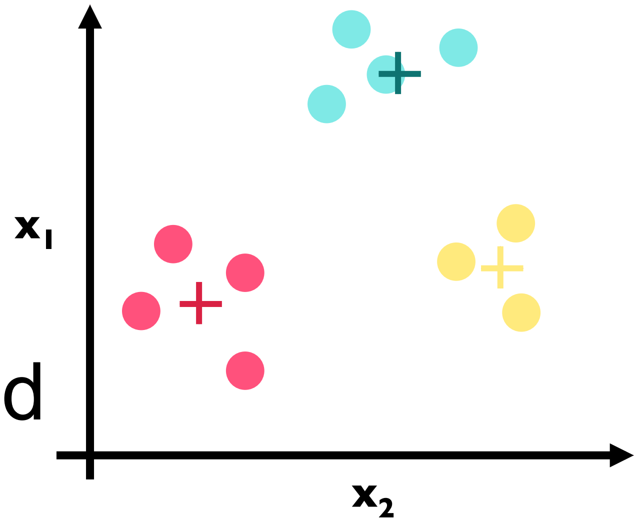


3) recalculate centroid positions

4) repeat steps 2-3 until stopping criteria met

1) choose k initial centroids (note that k is an input)

2) for each point:
   - find distance to each centroid
   - assign point to nearest centroid



3) recalculate centroid positions

4) repeat steps 2-3 until stopping criteria met

III. CLUSTER VALIDATION

In general, k-means will converge to a solution and return a partition of k clusters, even if no natural clusters exist in the data.

In general, k-means will converge to a solution and return a partition of k clusters, even if no natural clusters exist in the data.

We will look at two validation metrics useful for partitional clustering, cohesion and separation.

Cohesion measures clustering effectiveness within a cluster.

$$\hat{C}(C_i) = \sum_{x \in C_i} d(x, c_i)$$

Cohesion measures clustering effectiveness within a cluster.

$$\hat{C}(C_i) = \sum_{x \in C_i} d(x, c_i)$$

Separation measures clustering effectiveness between clusters.

$$\hat{S}(C_i, C_j) = d(c_i, c_j)$$

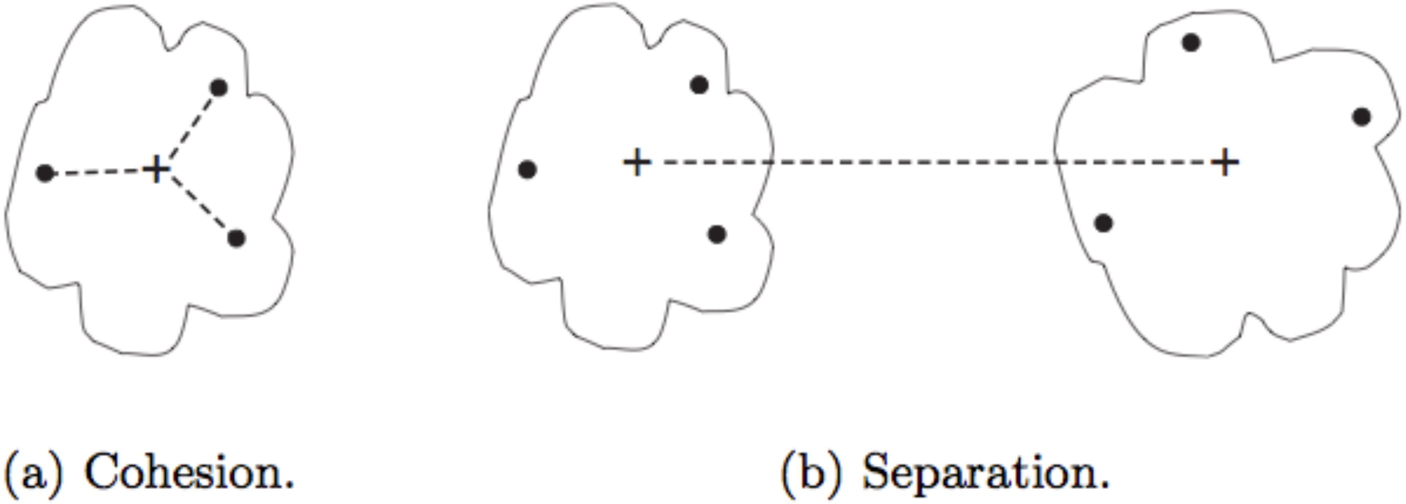


Figure 8.28. Prototype-based view of cluster cohesion and separation.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

One useful measure than combines the ideas of cohesion and separation is the silhouette coefficient. For point x_i, this is given by:

$$SC_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

such that:

a_i = average in-cluster distance to x_i
b_{ij} = average between-cluster distance to x_i
$b_i = min_j(b_{ij})$

The silhouette coefficient can take values between -1 and 1.

In general, we want separation to be high and cohesion to be low. This corresponds to a value of SC close to +1.

A negative silhouette coefficient means the cluster radius is larger than the space between clusters, and thus clusters overlap.

The silhouette coefficient for the cluster C_i is given by the average silhouette coefficient across all points in C_i:

$$SC(C_i) = \frac{1}{m_i} \sum_{x \in C_i} SC_i$$

The silhouette coefficient for the cluster C_i is given by the average silhouette coefficient across all points in C_i:

$$SC(C_i) = \frac{1}{m_i} \sum_{x \in C_i} SC_i$$

The overall silhouette coefficient is given by the average silhouette coefficient across all clusters:

$$SC_{total} = \frac{1}{k} \sum_{1}^{k} SC(C_i)$$

The silhouette coefficient for the cluster C_i is given by the average silhouette coefficient across all points in C_i:

$$SC(C_i) = \frac{1}{m_i} \sum_{x \in C_i} SC_i$$

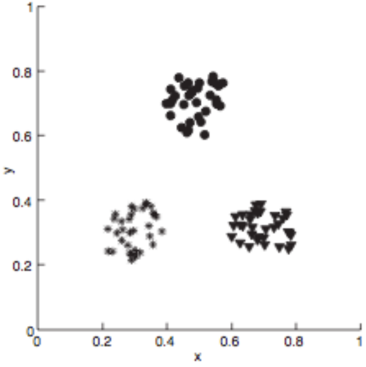
NOTE

This gives a summary measure of the overall clustering quality.

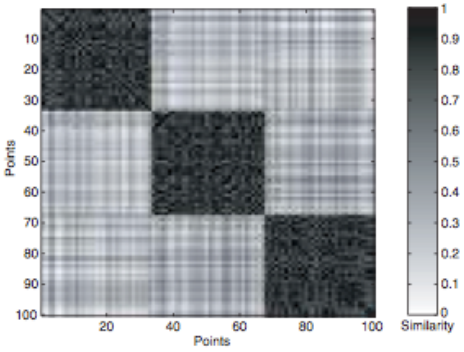
The overall silhouette coefficient is given by the average silhouette coefficient across all clusters:

$$SC_{total} = \frac{1}{k} \sum_{1}^{k} SC(C_i)$$

An alternative validation scheme is given by comparing the similarity matrix with an idealized (0/1) similarity matrix that represents the same clustering configuration.



(a) Well-separated clusters.



(b) Similarity matrix sorted by K-means cluster labels.

One useful application of cluster validation is to determine the best number of clusters for your dataset.

One useful application of cluster validation is to determine the best number of clusters for your dataset.

Q: How would you do this?

One useful application of cluster validation is to determine the best number of clusters for your dataset.

Q: How would you do this?
A: By computing the SSE or SC for different values of k.

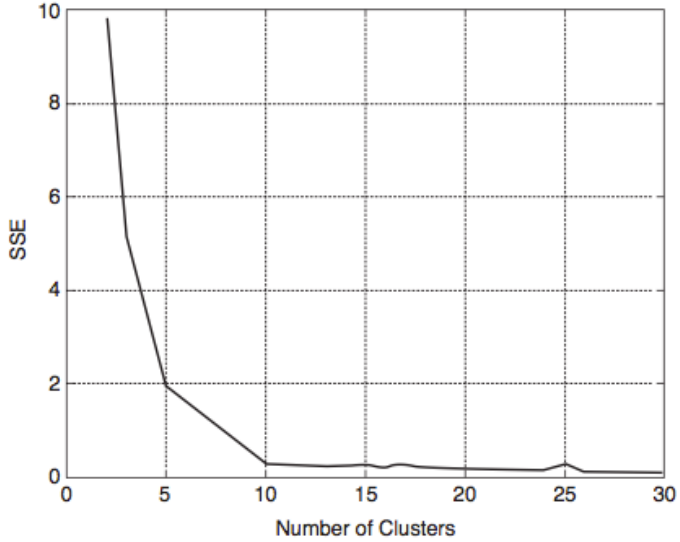


Figure 8.32. SSE versus number of clusters for the data of Figure 8.29.

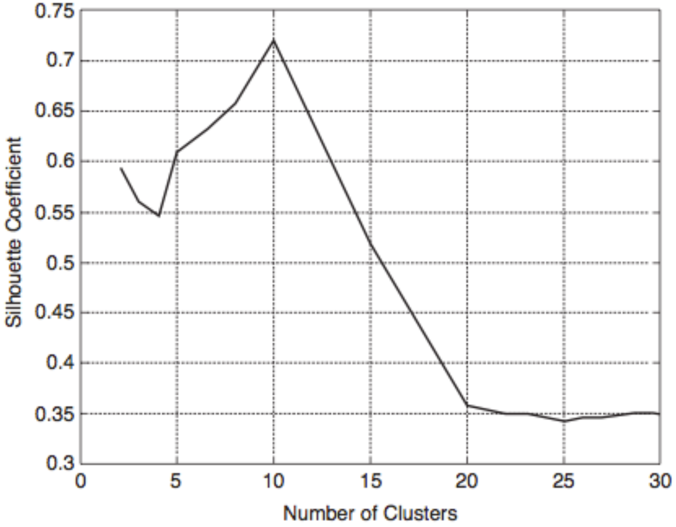


Figure 8.33. Average silhouette coefficient versus number of clusters for the data of Figure 8.29.

Ultimately, cluster validation and clustering in general are suggestive techniques that rely on human interpretation to be meaningful.

Strengths:

K-means is a popular algorithm because of its computational efficiency and simple and intuitive nature.

Strengths:
K-means is a popular algorithm because of its computational efficiency and simple and intuitive nature.

Weaknesses:
However, K-means is highly scale dependent, and is not suitable for data with widely varying shapes and densities.

EX: K-MEANS CLUSTERING