Real World Applications of Data Science

In partnership with:
Proscia Inc, Betamore, Spark B-more

Lecture 3: Topics in Unsupervised Learning Pt. 1

Topics in Unsupervised Learning

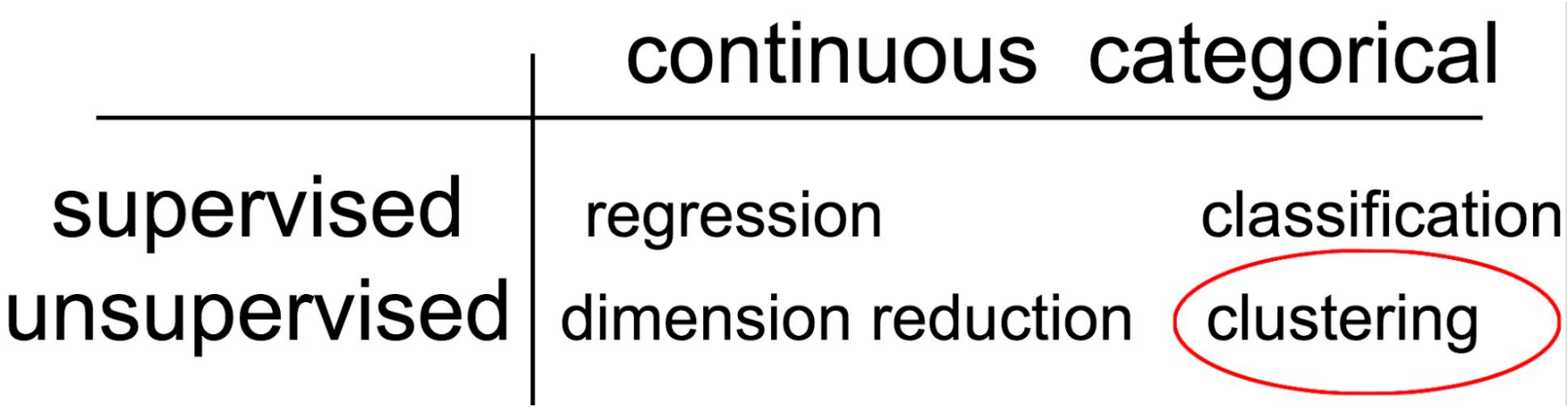
1) Clustering + Visualization
2) Principal Component Analysis
3) Dimensionality Reduction

Cluster analysis

Cluster Analysis

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

Cluster Analysis



Cluster Analysis

Q: What is a cluster?

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.

Cluster Analysis

Q: What is the purpose of cluster analysis?

Cluster Analysis

Q: What is the purpose of cluster analysis?

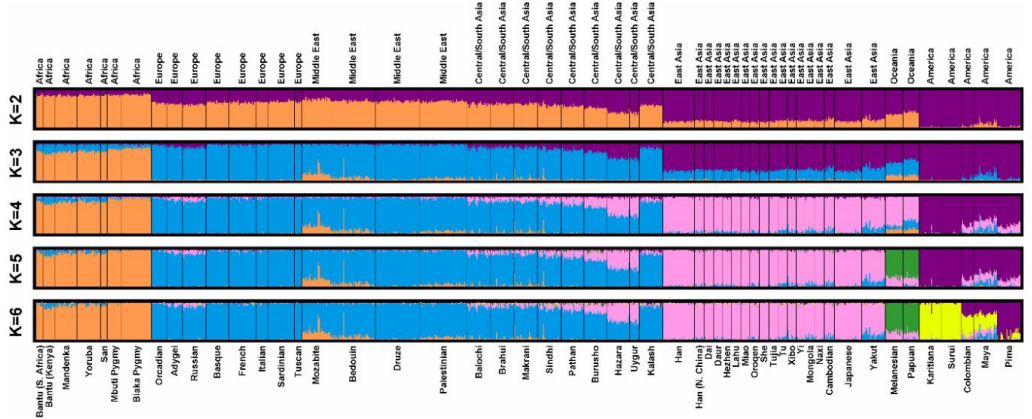
A: To enhance our understanding of a dataset by dividing the data into groups.

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

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

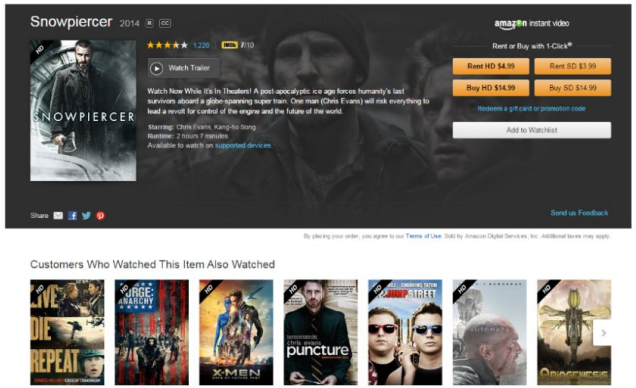
Cluster Analysis

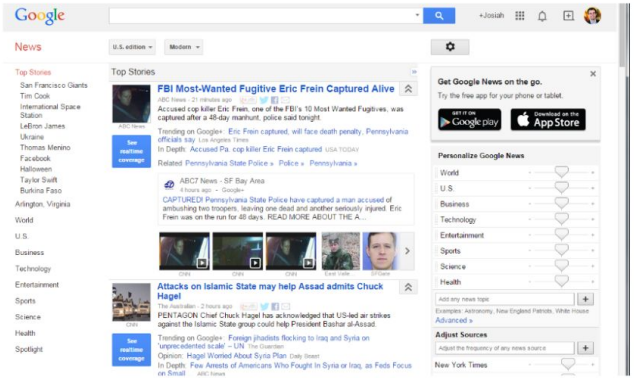
Clustering can be useful in a wide variety of domains, including **genetics**, consumer internet and business.



Cluster Analysis

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





Cluster Analysis

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

Intro K-means

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.

K- Means Clustering

K-means mechanics

Q: How does the algorithm work?

K-means mechanics

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

K-means mechanics

Q: How do you choose the initial centroid positions?

Choosing initial 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)

Assess similarity

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

Assess similarity

The similarity criterion is determined by the measure we choose.

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

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

Recomputing the center

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

Recomputing the center

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)

Convergence

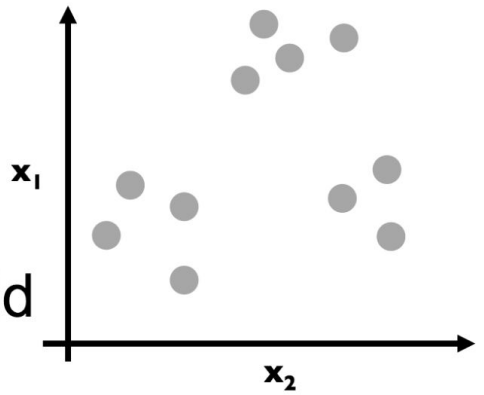
We iterate until some stopping criteria are met; in general, suitable convergence is achieved in a small number of steps.

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).

Basic k-means algorithm

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

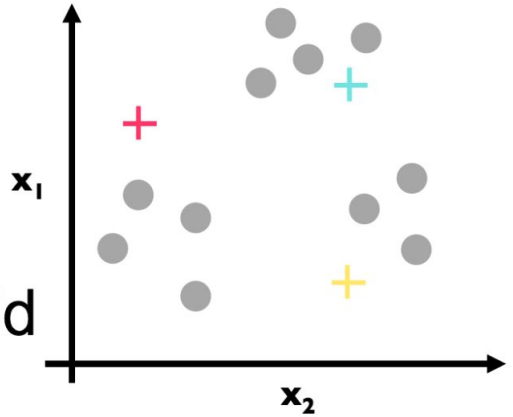


3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

Basic k-means algorithm

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

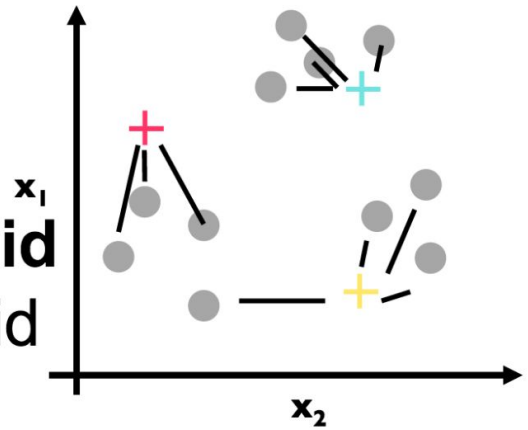


3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

Basic k-means algorithm

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

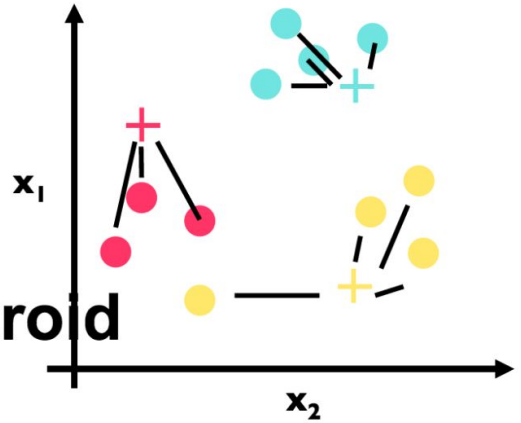


3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

Basic k-means algorithm

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

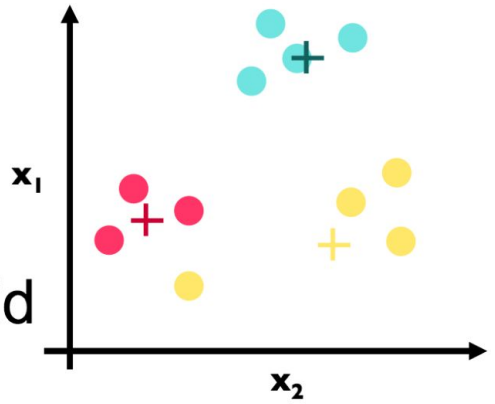


3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

Basic k-means algorithm

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



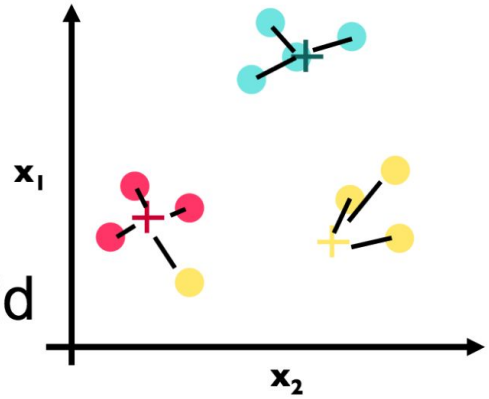
3) recalculate centroid positions

4) repeat steps 2-3 until stopping criteria met

Basic k-means algorithm

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



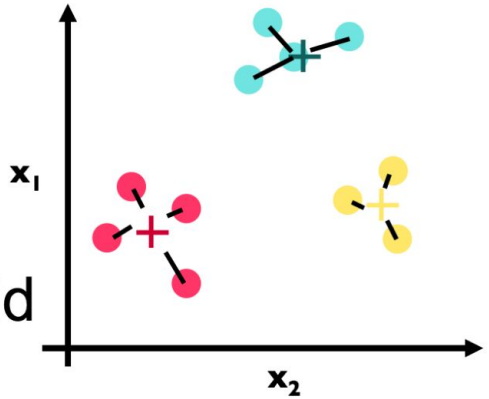
3) recalculate centroid positions

4) repeat steps 2-3 until stopping criteria met

Basic k-means algorithm

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



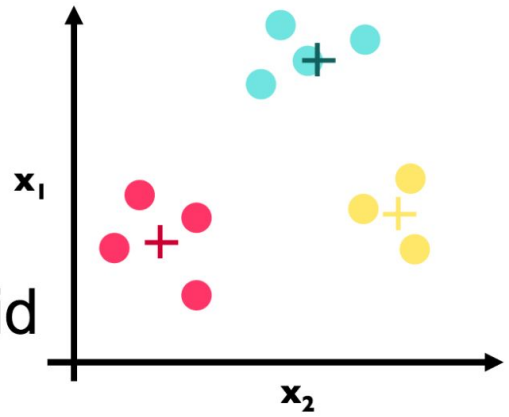
3) recalculate centroid positions

4) repeat steps 2-3 until stopping criteria met

Basic k-means algorithm

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

Cluster validation

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.

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

Cluster validation metrics

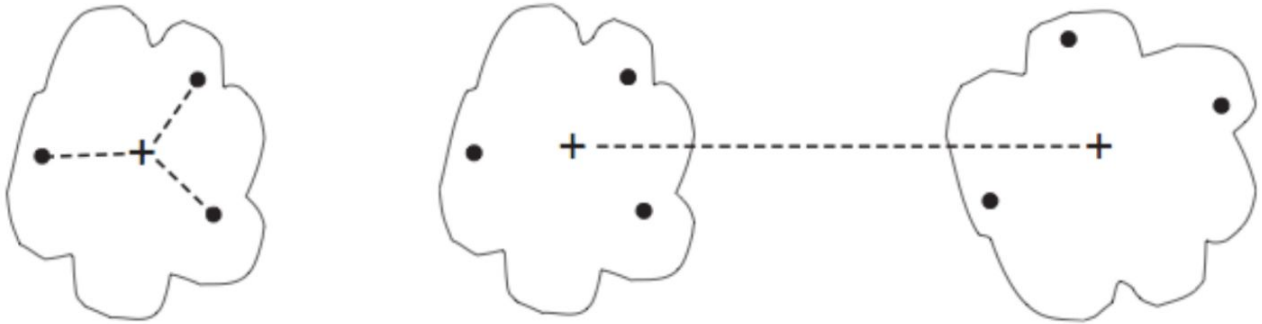
Cohesion measures clustering effectiveness within a cluster.

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

Separation measures clustering effectiveness between clusters.

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

Cluster validation metrics



(a) Cohesion.

(b) Separation.

Silhouette coefficient

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
- $b_i = min_j(b_{ij})$

Silhouette coefficient

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.