Unsupervised Learning
k-Means

Jayanth Rasamsetti
Founder www.sgmoid.com
Columbia University (MS)
IIT-Madras (B.Tech & M.Tech)

Agenda: Fundamentals of Unsupervised Learning

Clustering - Understanding Distance

Hierarchical clustering

K-Means and K-medoids

Why clustering and its applications

Why clustering?

1) To group similar objects/data points
2) To find homogeneous sets of customers
3) To segment the data in similar groups

Applications:

1) Marketing: Customer Segmentation & Profiling
2) Libraries: Book classification
3) Retail: Store Categorization

What is Clustering?

Clustering is a technique for finding similar groups in data, called clusters

Clustering is an Unsupervised Learning Technique

Clustering can also be thought of as a case reduction technique wherein it groups together similar records in cluster

What is a Cluster?

A cluster can be defined as a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters

How do we define “Similar” in clustering?

Based on Distance

Shoppers	Price Conscious	Brand Loyalty
A	2	4
B	8	2
C	9	3
D	1	5
E	8	1

How do we define “(dis) Similar” ?

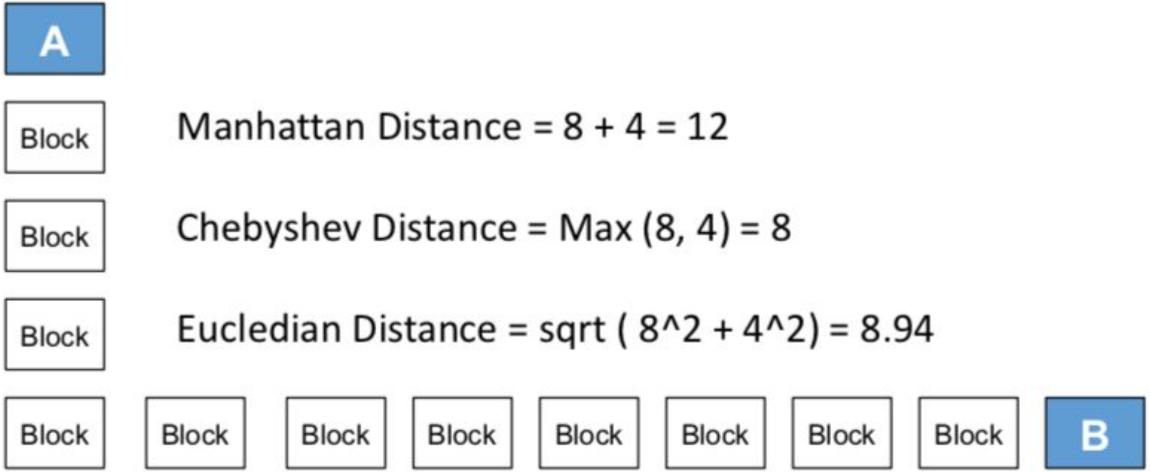
Similar in clustering is based on Distance

Various distance measures

Euclidean Distance

Chebyshev Distance

Manhattan Distance ...and more



Chebyshev Distance

In mathematics, Chebyshev distance is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension

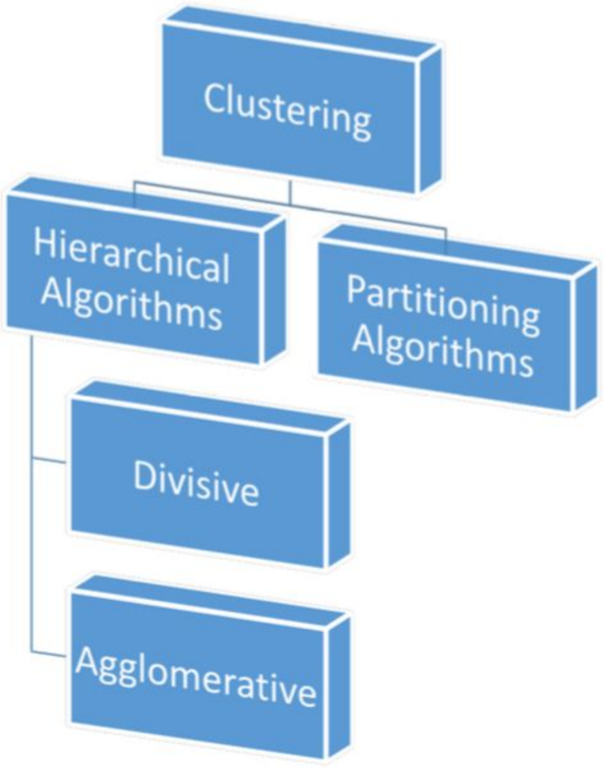
Assume two vectors: A (x1, y1, ... z1) & B (x2, y2, ... z2)

Chebyshev Distance = Max (| x2 - x1 | , | y2 – y1 | , ... | z2 – z1 |)

Application: Survey / Research Data where the responses are Ordinal Reference

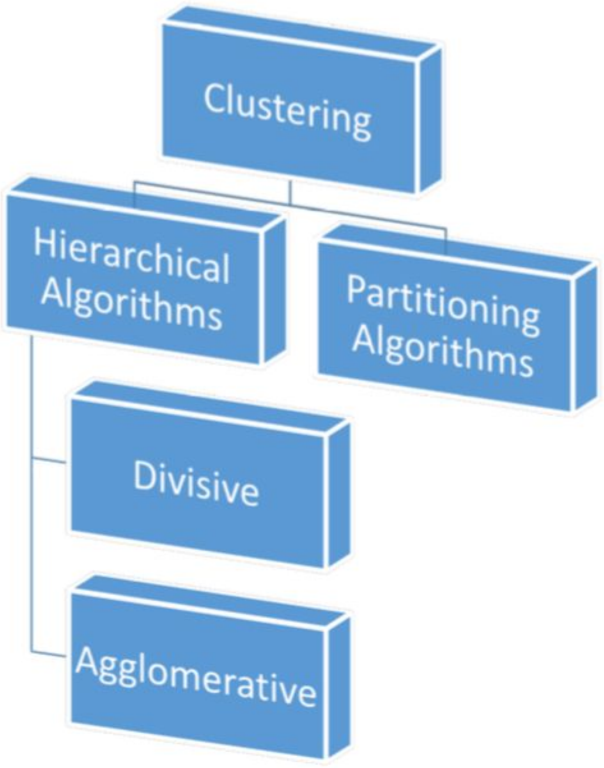
Link: https://en.wikipedia.org/wiki/Chebyshev_distance

Types of Clustering Procedures



- Hierarchical clustering is characterized by a tree like structure and uses distance as a measure of (dis)similarity
- Partitioning Algorithms starts with a set of partitions as clusters and iteratively refines the partitions to form stable clusters

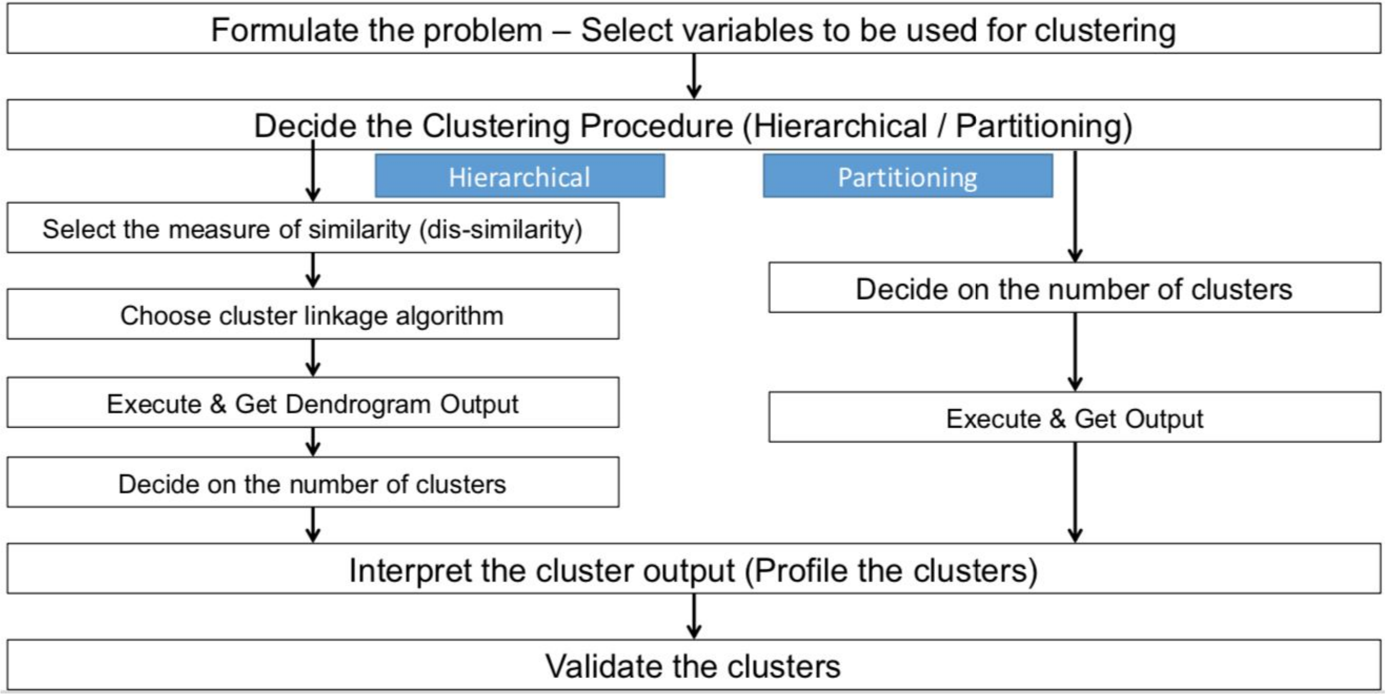
Types of Clustering Procedures



Agglomerative: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

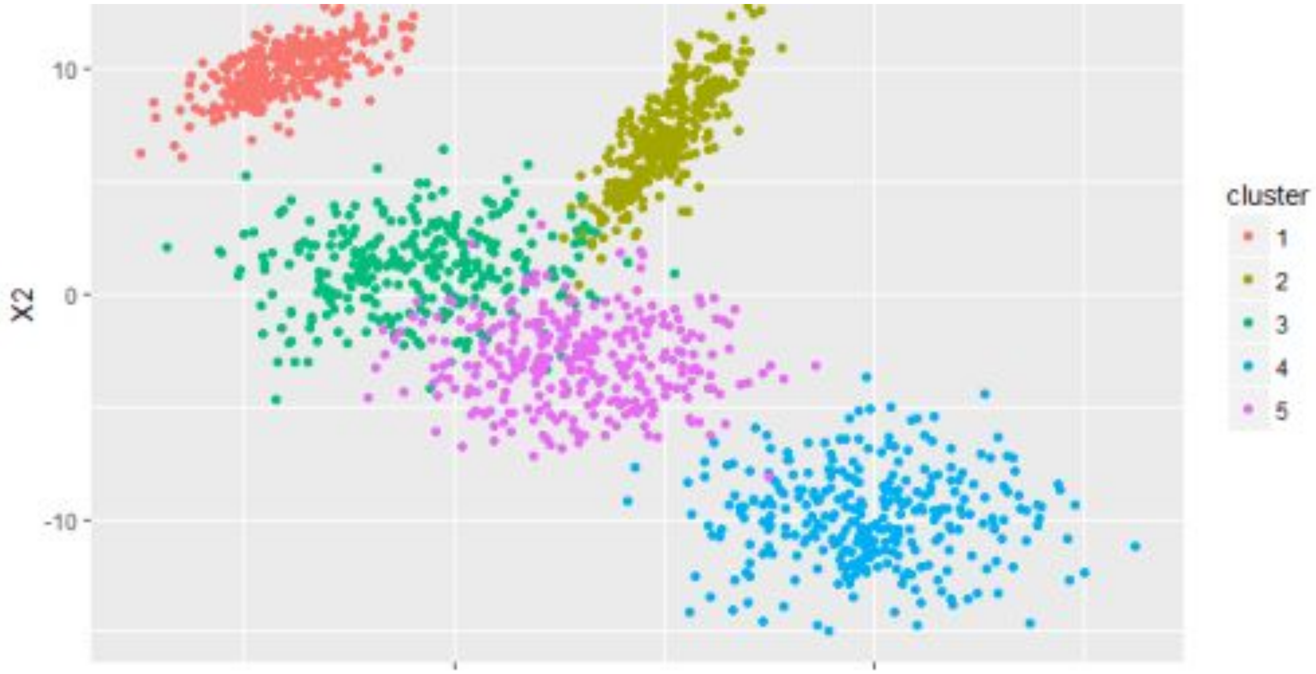
Divisive: This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Steps involved in Clustering



Partitioning Clustering

K Means Clustering



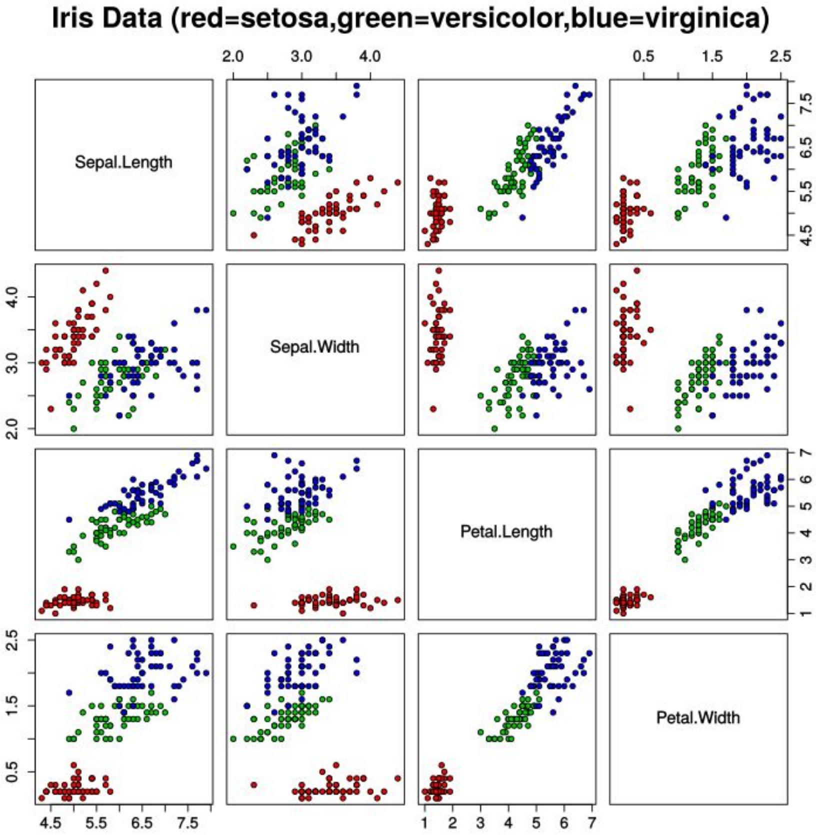
k-Means Clustering

K-Means is the most used, non-hierarchical clustering technique

It is not based on Distance

It is based on within cluster Variation, in other words Squared Distance from the Centre of the Cluster

The algorithm aims at segmenting data such that within cluster variation is reduced (WSS)



k-Means Algorithm

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
```

Steps

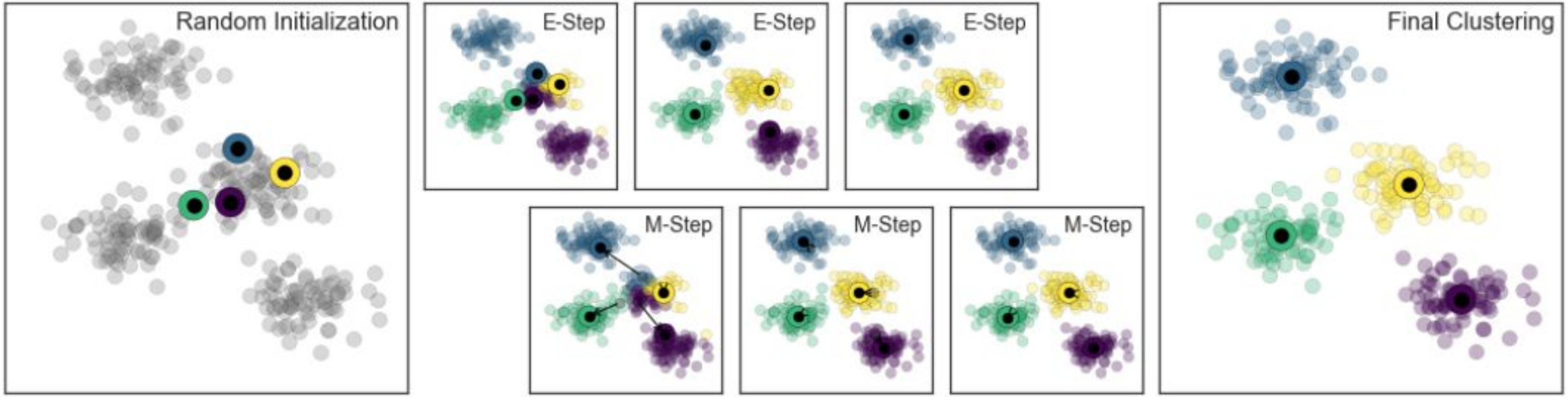
1. Assume K Centroids (for K Clusters)
2. Compute Euclidean distance of each objects with these Centroids
3. Assign the objects to clusters with shortest distance
4. Compute the new centroid (mean) of each cluster based on the objects assigned to each clusters. The K number of means obtained will become the new centroids for each cluster
5. Repeat step 2 to 4 till there is convergence
   a) i.e. there is no movement of objects from one cluster to another
   b) Or threshold number of iterations have occurred

Also called Expectation Maximization!

k-Means Algorithm (Expectation Maximization)

E-Step: assign points to the nearest cluster center

M-Step: set the cluster centers to the mean



k-Means advantages

K-means is superior technique compared to Hierarchical technique as it is less impacted by outliers

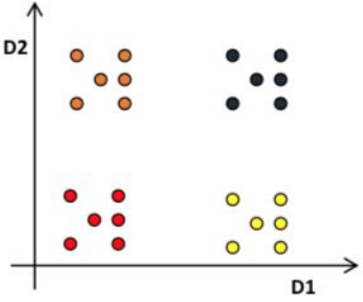
Computationally it is more faster compared to Hierarchical

Preferable to use on interval or ratio-scaled data as it uses Euclidean distance... desirable to avoid using on ordinal data

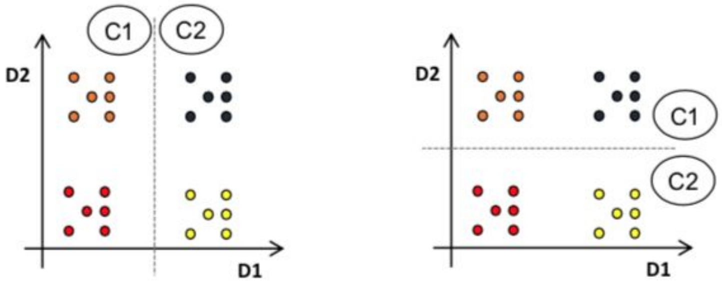
Challenge – Number of clusters are to be predefined and to be provided as input to the process

Why find optimal No. of Clusters?

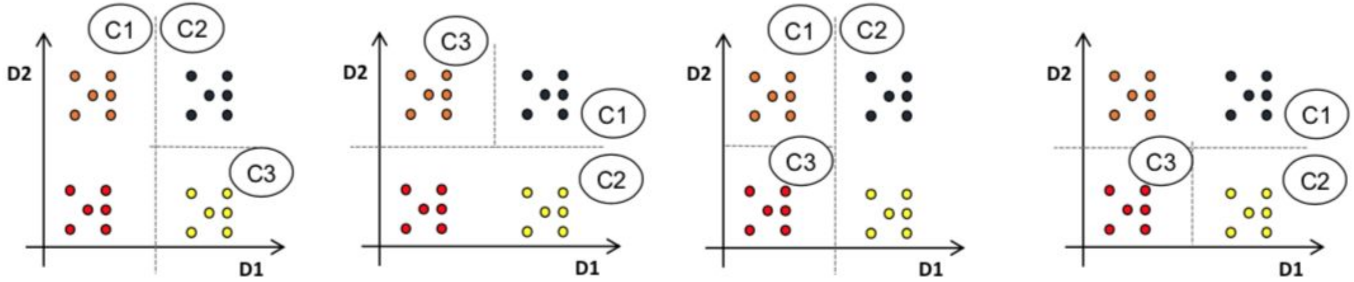
Data to be clustered



- Two Clusters – 2 possible solution



- Three Clusters – Multiple possible solution



Retail Customer Data Case Study

As a Machine Learning Engineer, you are asked to find out the patterns in the data using any unsupervised techniques

	Cust_ID	Name	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
0	1	A	10000	2	1	1	0
1	2	B	7000	3	0	10	9
2	3	C	7000	7	1	3	4
3	4	D	6500	5	1	1	4
4	5	E	6000	6	0	12	3
5	6	F	4000	3	0	1	8
6	7	G	2500	5	0	11	2
7	8	H	2500	3	0	1	1
8	9	I	2000	2	0	2	2
9	10	J	1000	4	0	1	7

WSS of Clusters (In Class Exercise: 10min Python)

```python
## Identify the optimal number of clusters
# elbow method
cluster_range = range( 1, 10 )
cluster_wss = []

for num_clusters in cluster_range:
    clusters = KMeans( num_clusters )
    clusters.fit(scaled_RCDF)
    cluster_wss.append( clusters.inertia_ )
from collections import OrderedDict
clusters_df = pd.DataFrame( OrderedDict (
        {"num_clusters": cluster_range,
        "cluster_wss": cluster_wss }
        ) )
clusters_df[0:10]
```

number of clusters

number of cases

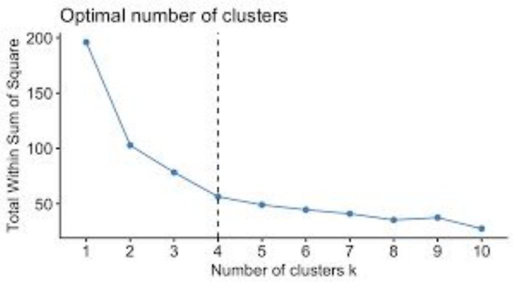
centroid for cluster *j*

case *i*

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

Optimal No. of Clusters: (In Class Exercise: 10min Python)

```
plt.figure(figsize=(12,6))
plt.xlabel('Number of Clusters')
plt.ylabel('Within Sum of Squares')
plt.xticks(np.arange(min(clusters_df.num_clusters),
                     max(clusters_df.num_clusters)+1,
                     1.0))
plt.plot( clusters_df.num_clusters,
         clusters_df.cluster_wss,
         marker = "o" )
```



Profiling the Clusters: (In Class Exercise: 10 min Python)

```
## Profiling the clusters

clusterer = KMeans(n_clusters=2, random_state=10)
cluster_labels = clusterer.fit_predict(scaled_RCDF)
cluster_labels
KRCDF['Clusters'] = cluster_labels

clus_profile = KRCDF.iloc[:,2:8].groupby(['Clusters'],
                                         as_index=False).mean()
clus_profile
```

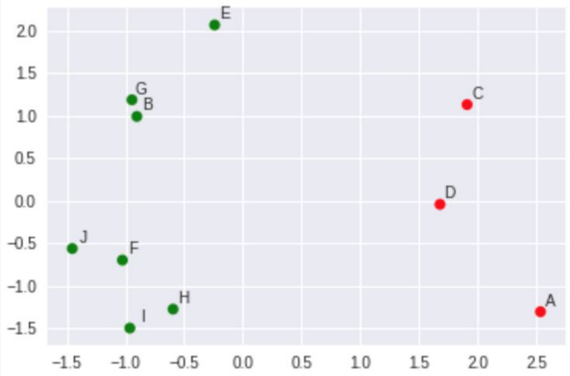
	Clusters	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
0	0	7833.333333	4.666667	1.0	1.666667	2.666667
1	1	3571.428571	3.714286	0.0	5.428571	4.571429

View the Points: (In Class Exercise: 10 min Python)

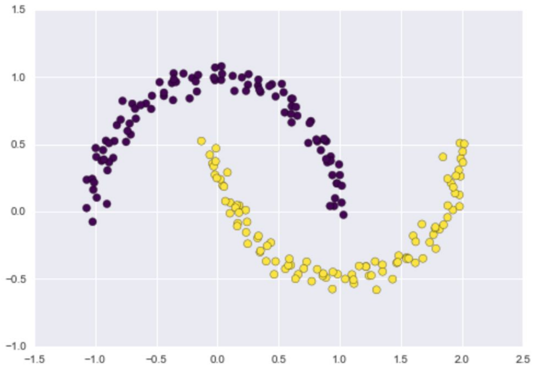
```
## Show the Cluster Plot
plt.scatter(x=plot_columns[:,0],
            y=plot_columns[:,1],
            c=KRCDF['color'].values.tolist(),
            s=50, edgecolors='none')

for label, x, y in zip(
        plot_labels, plot_columns[:,0],
        plot_columns[:,1]) :
    plt.annotate(
    label,
    xy=(x, y), xytext=(10, 2),
    textcoords='offset points', ha='right', va='bottom',
    )
    plt.xlabel('PC1')
    plt.ylabel('PC2')

plt.show()
```



k-Means Limitations



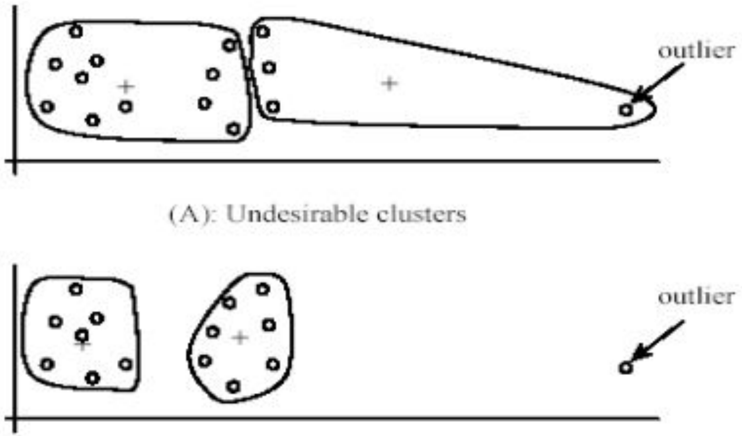
1. What is clusters are not clearly separable?

2. Applicable to data where “mean” is well-defined
Restricts applicability to only Euclidean spaces i.e. if categorical attributes present (e.g. Marital Status, Gender), “mean” not meaningful

K-Medoids (a minor variant): Choose most centrally located point within the cluster as representative (This step is more computationally intensive)

Sensitivity to outliers in data – Detect and remove outliers before clustering – K-Medians is relatively more robust to outliers

k-Means Limitations



(A): Undesirable clusters

(B): Ideal clusters

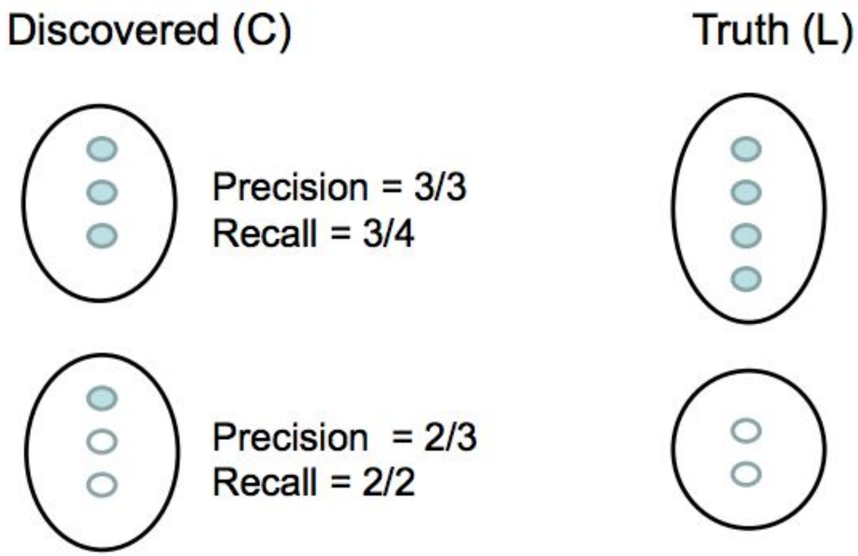
Sensitivity to outliers in data
Detect and remove outliers before clustering

K-Medians is relatively more robust to outliers and because a medoid is less influenced by outliers or other extreme values than a mean

Works efficiently for small data sets but does not scale well for large data sets

Evaluation of Cluster Quality

Extrinsic Evaluation – Given Ground Truth Data



$$AlgoPrecision = \sum_i \frac{|C_i|}{n} \max_j Precision(C_i, L_j)$$

$$Precision\ (C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$$

$$AlgoRecall = \sum_i \frac{|L_i|}{n} \max_j Recall\ (C_j, L_i)$$

$$Recall\ (C_j, L_i) = \frac{|C_j \cap L_i|}{|L_i|}$$

$$F = \sum_i \frac{|L_i|}{n} \max_j \{F(C_j, L_i)\}$$

$$F(C_j, L_i) = \frac{2 \times Recall(C_j, L_i) \times Precision(C_i, L_j)}{Recall(C_j, L_i) + Precision(C_i, L_j)}$$

Evaluation of Cluster Quality

Intrinsic Evaluation

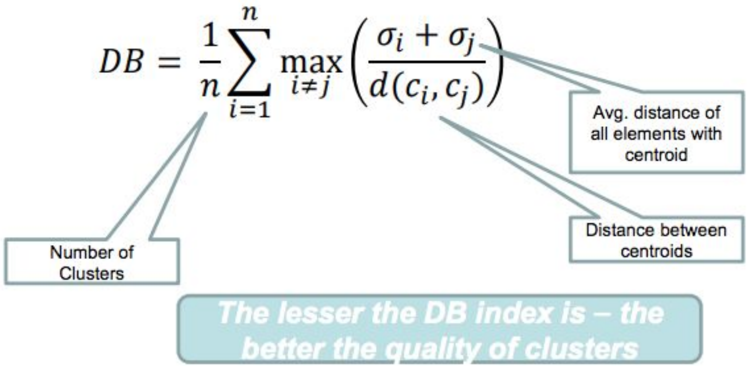
When no gold standard data is available

Develop measures for some general goodness criterion

E.g. Good clusters should high intra-cluster similarity and low inter cluster similarity Davies-Bouldin (DB) Index

To check the stability of the clusters take a random sample of 95% of records. Compute the clusters. If the clusters formed are very similar to the original, then the clusters are fine

$$DB = \frac{1}{n}\sum_{i=1}^{n} \max_{i \neq j}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$



Next steps after clustering

Clustering provides you with clusters in the given dataset

Clustering does not provide you rules to classify future records

To be able to classify future records you may do the following

Build Discriminant Model on Clustered Data

Build Classification Tree Model on Clustered Data

https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

k-Means versus Hierarchical

K-means produces a single partitioning

K-means needs the number of clusters to be specified

K-means is usually more efficient run-time wise

Hierarchical Clustering can give different partitions depending on the level-of-resolution we are looking at

Hierarchical clustering doesn’t need the number of clusters to be specified

Hierarchical clustering can be slow (has to make several merge/Split decisions)

Clustering applications - Astrostatistics

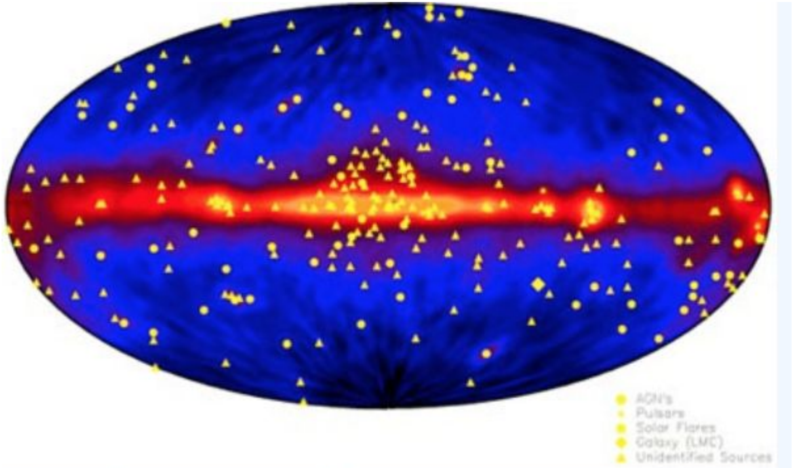


Image: `http://science.hq.nasa.gov`

THREE TYPES OF GAMMA-RAY BURSTS

SOMA MUKHERJEE,[1,2,3] ERIC D. FEIGELSON,[4] GUTTI JOGESH BABU,[5] FIONN MURTAGH,[6,7]
CHRIS FRALEY,[8] AND ADRIAN RAFTERY[8]

Received 1998 February 9; accepted 1998 June 25

“One” schematic for addressing problems in machine learning...

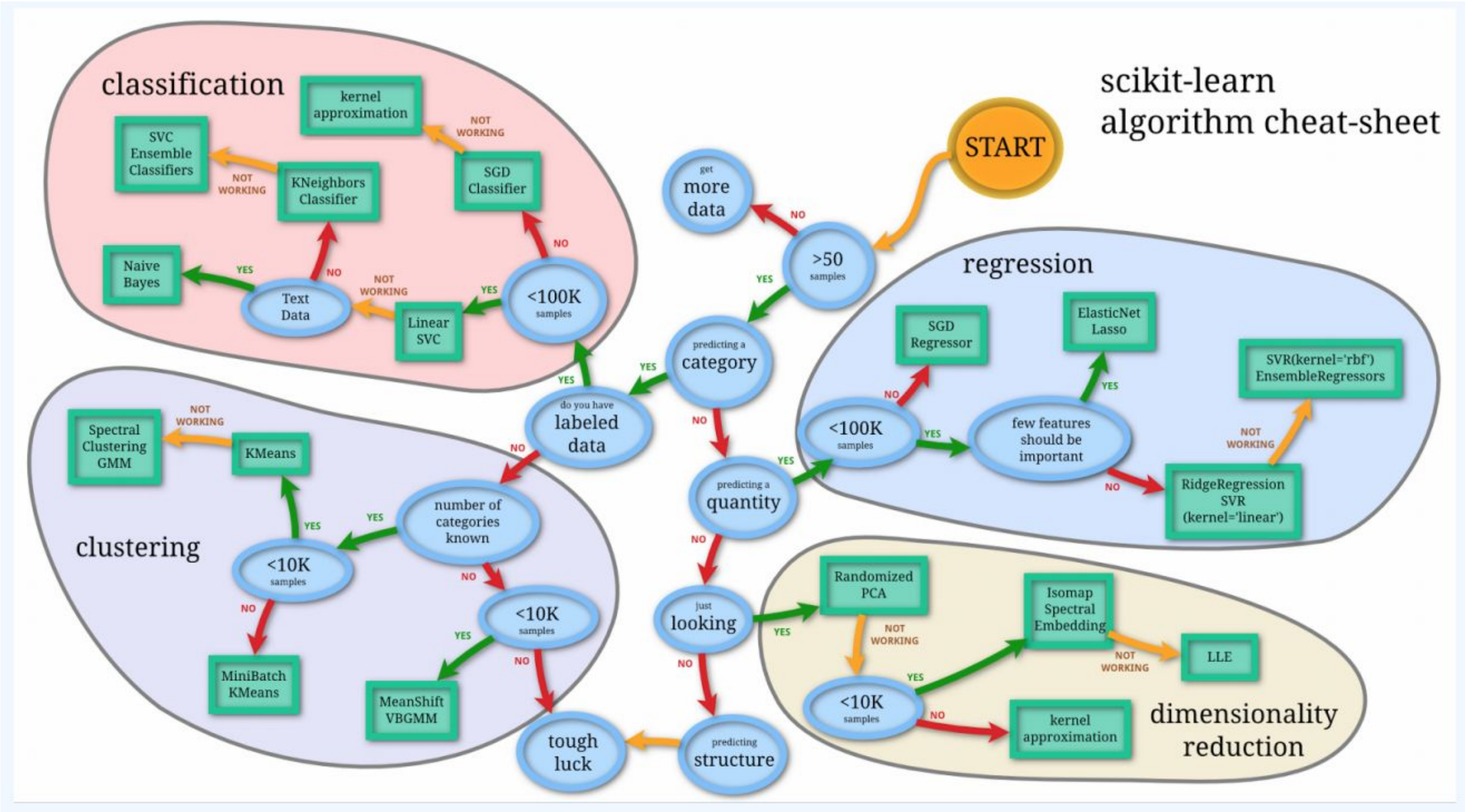


Image: Andy’s Computer Vision and Machine Learning Blog http://peekaboo-vision.blogspot.com

Summary

Introduced the paradigm of “Unsupervised Learning” – The task of discovering intrinsic patterns from data without any supervision

Depending on the specific objective to be optimized and assumptions made about data, there are many clustering algorithms proposed in literature

Some clustering algorithms we discussed:

Agglomerative Clustering - Case Study & Worked out Example
K-Means - Worked out Example

Practical issues while using the above algorithms. We also studied the notion of cluster evaluation

References

References

Chapter 9: Cluster Analysis (http://www.springer.com)
Google search : “www.springer.com cluster analysis chapter 9”

http://sites.stat.psu.edu/~ajw13/stat505/fa06/19_cluster/09_cluster_wards.html •

https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/

Thank you! Questions