Machine Learning with Python

Session 12: Cluster Analysis

Arghya Ray

Identifying Similarities in Data

- Large amount of information are constantly being generated, organized, analyzed and stored.
- Identifying important patterns, associations and groupings of similar data can be helpful for customers as well as organizations.
- **Data Clustering** can help us make sense of the huge amount of data by discovering hidden groupings of similar items.
- Clustering can help in analysis of online social networks.
- Clustering can help to distinguish between different items. E.g. Fresh vegetables are more similar to each other than frozen items.
- Clustering is useful in ***market segmentation*** by partitioning the target market data into groups such as customer who share the same interests or those with common needs. Identifying clusters of similar items can help develop a marketing strategy that addresses the needs of specific clusters.
- Data Clustering can help ***to identify, learn, or predict the nature of new data items***– especially how new data can be linked with making predictions. For e.g., in pattern recognition analyzing patterns in the data (such as buying patterns in particular regions or age groups) can help to develop predictive analytics to predict the nature of future data items that can fit well with established patterns.
- Clustering can help in ***dividing the e-mail dataset into spam and non-spam messages***.
- Data Clustering is also helpful ***in image segmentation*** for analyzing the image more easily.
- Data Clustering can help ***in information retrieval from a collection of data***, mainly documents (using tf-idf concepts).
- On the other hand, finding important **association rules in a dataset** of customer transactions helps a company to maximize revenue by deciding which products should be on sale, how to position products in the store's aisles, and how and when to offer promotional pricing.

Types of Cluster Analysis Methods:

- **Partitional Methods**: Partitional methods obtain a single level partition of objects. These methods are usually based on a greedy heuristics that are used to obtain a local optimum solution. Given *n* objects, these methods make k<=n clusters.
  - ***K-means***: Each of the K-clusters is represented by the mean of the objects inside each cluster.
  - ***Density-Based***: It is based on the assumption that clusters have high density collection of data of arbitrary shape that are separated by a large space of low density data (which is assumed to be the noise).
  - ***Expectation-Maximization***: The EM method assigns objects to different clusters with certain probabilities in an attempt to maximize the expectation (or likelihood) of assignment.
- **Hierarchical Methods**: Hierarchical methods obtain a nested partition of the objects resulting in a tree of clusters.
  - ***Agglomerative***: Start with each object in an individual cluster and then try to merge similar clusters into larger and larger clusters.
  - ***Divisive***: Start with one cluster and then split into smaller and smaller clusters.
- **Grid Based method**: The object space rather than the data is divided into a grid. Grid partitioning is based on characteristics of the data and such methods can deal with non-parametric data more easily.
- **Model based method**: A model is assumed based on a probability distribution. Essentially the algorithm tries to build clusters with a high level of similarity with them and a low level of similarity between them. It tries to minimise the squared-error function.

Requirements of Clustering in Data Mining

The following points throw light on why clustering is required in data mining –

Scalability – We need highly scalable clustering algorithms to deal with large databases.

Ability to deal with different kinds of attributes – Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.

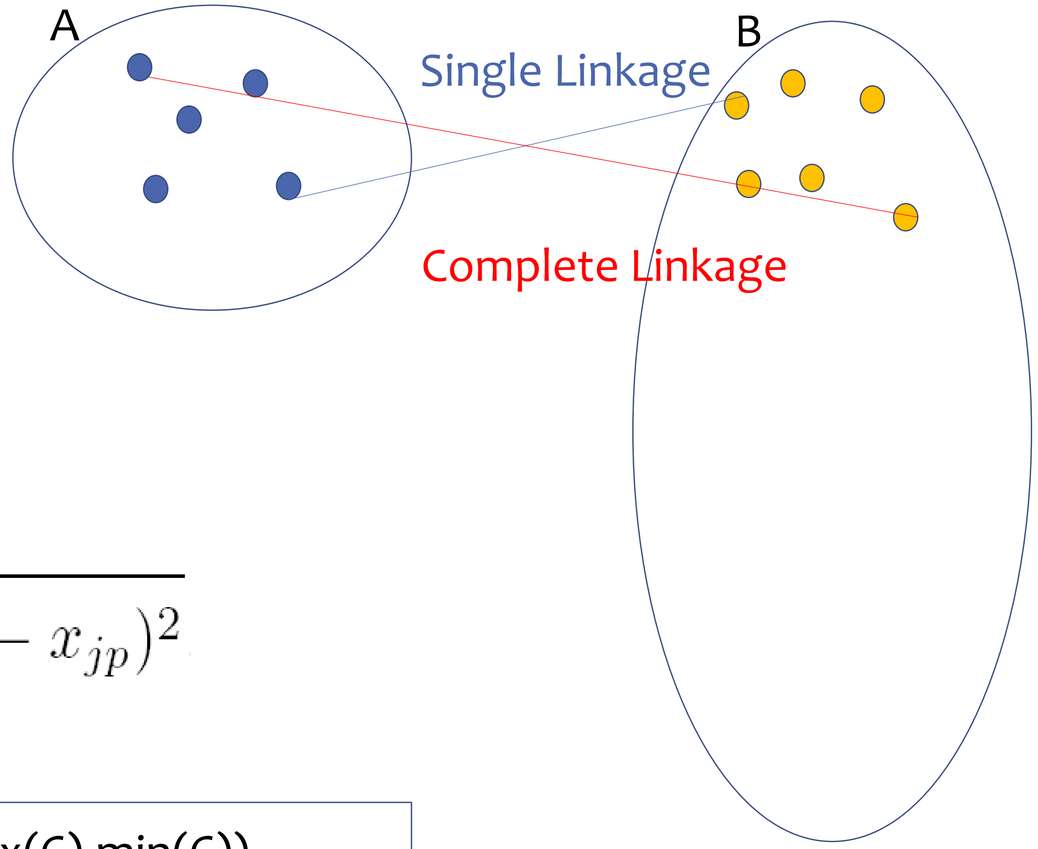
Discovery of clusters with attribute shape – The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.

High dimensionality – The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.

Ability to deal with noisy data – Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.

Interpretability – The clustering results should be interpretable, comprehensible, and usable.

Some important concepts:



- **Measuring Distance**
  - Between records: Distance between each record in a cluster.
  - Between clusters: Distance between each cluster.
- **Distance Between Two Records:** Euclidian distance is most popular.

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

- **Normalizing:**

(x-min(C))/(max(C)-min(C))

  - ***Problem***: Raw distance measures are highly influenced by scale of measurements
  - ***Solution***: Normalize (standardize) the data first.
  - Subtract mean, divide by std. deviation. (Also called z-scores).
  - Example: For 22 utilities, Avg. sales = 8,914; Std. dev. = 3,550. Hence, Normalized score: (9,077-8,914)/3,550 = 0.046

Measuring Distance Between Clusters:

- **Single Linkage**
  - Minimum Distance (Cluster A to Cluster B)
  - Distance between two clusters is the distance between the pair of records Ai and Bj that are closest.
- **Complete Linkage**
  - Maximum Distance (Cluster A to Cluster B)
  - Distance between two clusters is the distance between the pair of records Ai and Bj that are farthest from each other
- **Average Linkage**
  - Distance between two clusters is the average of all possible pair-wise distances
- **Centroid**
  - Distance between two clusters is the distance between the two cluster centroids.
  - Centroid is the vector of variable averages for all records in a cluster

K-Means Clustering:

- The K-means method may be described as follows:
  1. Select the number of clusters. Let this be ***k***.
  2. Pick k seeds as centroids of the ***k*** clusters. The seeds may be picked randomly unless the user has some insight about the data.
  3. Compute the Euclidean distance of each object in the dataset from each of the centroids.
  4. Allocate each object to the cluster it is nearest to based on the distances computed in the previous step.
  5. Compute the centroids of the clusters by computing the means of the attribute values of the objects in each cluster.
  6. Check if the stopping criteria has been met. If yes, stop. If not, go to step 3.
- The ***k-means method*** uses the Euclidean distance method, which appears to work well with compact clusters.
- If the Manhattan distance is used the method is called ***k-median method***. This method may be less sensitive to outliers.
- K-means Algorithm: Choosing k and Initial Partitioning

  Manhattan Distance-> $|x1-x2| + |y1-y2|$
  - Choose k based on the how results will be used. E.g., "How many market segments do we want?"
  - Also experiment with slightly different k's.
  - Initial partition into clusters can be random, or based on domain knowledge. If random partition, repeat the process with different random partitions
- For clustering to be effective all attributes should be converted to a similar scale unless you want to give more weight to some attributes that are relatively large in scale.

1 6 3 7 4 9

Step 1: K= 2

Step 2: C1: 1 6 3 C2: 7 4 9
Mean 3.333 6.667

Step 3: **C1:** **1** 6 3
M1: 2.3333 2.667 0.3333
M2: 5.667 0.667 3.667

C2: **7** **4** **9**
M2: 1.667 2.667 2.333
M1: 4.337 0.7 5.7

Step 4: 1 → C1; 6→C2; 3→C 1 ; 7→C 2 ; 4→C 1 ; 9→C2

Step 5: C1: 1 3 4 C2: 6 7 9
Mean 2.667 7.333

Step 6: C1: 1 3 4 C2: 6 7 9
M1: 1.667 0.333 1.33 M1: 3.333 4.333 6.333
M2: 6.333 4.333 3.333 M2: 1.333 0.333 1.67

Step 7: 1→ C1; 3→C1; 4→ C1 6→ C2; 7→ C2; 9→C2

Q1. Use k-mean clustering to divide the following set of numbers into two clusters.

1 2 3 4 5 6 7 8 9 10

Q2. Use k-median method to form clusters of students based on the data given below:

Student	Age	Mark1	Mark2	Mark3
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52
S4	20	55	55	55
S5	22	85	86	87
S6	19	91	90	89
S7	20	70	65	60
S8	21	53	56	59
S9	19	82	82	60
S10	47	75	76	77

K=3

Steps 1 and 2: Let the three seeds be the first three students as shown.

Student	Age	Mark1	Mark2	Mark3
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52

Step 3 and 4: Compute the distances using the four attributes and using the sum of absolute differences (k-Median method)

	Age	Mark1	Mark2	Mark3	Distance from Clusters			Allocation to the nearest Cluster
C1	18	73	75	57	From C1	From C2	From C3	
C2	18	79	85	75				
C3	23	70	70	52				
S1	18	73	75	57	0	34	18	C1
S2	18	79	85	75	34	0	52	C2
S3	23	70	70	52	18	52	0	C3
S4	20	55	55	55	42	76	36	C3
S5	22	85	86	87	57	23	67	C2
S6	19	91	90	89	66	32	82	C2
S7	20	70	65	60	18	46	16	C3
S8	21	53	56	59	44	74	40	C3
S9	19	82	82	60	20	22	36	C1
S10	47	75	76	77	52	44	60	C2

Step 5: The new cluster means of clusters are given in the table below:

Manhattan Distance-> $|x1-x2| + |y1-y2|+|z1-z2|+|w1-w2|$

Student	Age	Mark1	Mark2	Mark3
C1	18.5	77.5	78.5	58.5
C2	26.5	82.5	84.3	82.0
C3	21	61.5	61.5	65.5

Step 3 and 4: Using this new cluster means compute the distances of each object to each of the means and allocate to nearest cluster.

	Age	Mark1	Mark2	Mark3	Distance from Clusters			Allocation to the nearest Cluster
C1	18.5	77.5	78.5	58.5	From C1	From C2	From C3	
C2	26.5	82.5	84.3	82.0				
C3	21	61.5	61.5	65.5				
S1	18	73	75	57	10	52.3	28	C1
S2	18	79	85	75	25	19.8	62	C2
S3	23	70	70	52	27	60.3	23	C3
S4	20	55	55	55	51	90.3	16	C3
S5	22	85	86	87	47	13.8	79	C2
S6	19	91	90	89	56	28.8	92	C2
S7	20	70	65	60	24	60.3	16	C3
S8	21	53	56	59	50	86.3	17	C3
S9	19	82	82	60	10	32.3	46	C1
S10	47	75	76	77	52	41.3	74	C2

Step 6: The clusters have not changed and hence we can stop.
Therefore, **Cluster 1:** S1, S9. **Cluster 2:** S2, S5, S6, S10. **Cluster 3:** S3, S4, S7, S8.

Cluster	C1	C2	C3
C1	5.9	26.5	23.3
C2	29.5	14.3	42.6
C3	23.9	41.0	10.7

Within cluster and between cluster distances

The content of the slides are prepared from different textbooks.

References:

- Data Mining and Predictive Analytics, By Daniel T. Larose. Copyright 2015 John Wiley & Sons, Inc.
- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..