Quiz 5 Unsupervised Learning

1. **Summarize a use case where you must use Manhattan Distance as the similarity measure in unsupervised learning.**

The use of any particular distance/similarity metric has a dependence on the kind of coordinate system used on the dataset at hand. Some examples of this are when analyzing chess data, for this case the Manhattan distance might be a better similarity metric than the Euclidean distance. Another example is when analyzing the distance between houses in a neighborhood.

Another important moment when Manhattan Distance might yield better results is if the input variable are not in a similar type (age, gender, height, etc.). This as the Euclidean distance becomes a poor choice as the number of this different type dimensions increases (course of dimensionality).

In summary the Manhattan Distance tend to outperform Euclidean distance for datasets arranged in a grid and in problems where high dimensionality is present and the different features represent different types of data. There is a great research paper where there is a discussion on the importance of choosing the right distance metric especially for unsupervised learning settings, you can find the article in the following link:

[*https://bib.dbvis.de/uploadedFiles/155.pdf*](https://bib.dbvis.de/uploadedFiles/155.pdf)

1. **Summarize (1 paragraph each) any 5 unsupervised learning algorithms, how they work and what they are used for.**

**K-Means:**

This might be the central algorithm in unsupervised learning operations. This algorithm defines the features present in a dataset and performs a grouping into clusters based on common elements. In this method you need to cluster the data points into k groups. The result from the algorithm are different clusters each with its own label. Each group is defined by calculating a centroid for each k label and every point is compared and assign to a particular cluster based on its similarity to any of the existing clusters. Some examples of the used of the algorithm are: audience segmentation, anomaly detection, patter recognition or even inventory management.

**Hierarchical clustering:**

Hierarchical clustering is an algorithm which builds a hierarchy of clusters. It begins with all the data which is assigned to a cluster of their own. Here, two close cluster are going to be in the same cluster. This algorithm ends when there is only one cluster left. There are two main approaches to this algorithm: agglomerative or divisive. In the first one each individual point is initially consider a cluster and at each iteration the clusters merge with others until a K cluster is form (which is defined by the user). The second approach assumes that all data points are one big cluster and after each iteration we obtain smaller clusters until the K cluster is reach (same as before, this is defined by the user). One of the uses of this technique is to chart evolution through phylogenetic trees to find the answer on how certain species relate to each other.

**PCA:**

Principal Component Analysis is a well-known dimensionality reduction algorithm used for multiple reasons, such as: data visualization and to help to define new features (linear or nonlinear combination of given features) that contain most of the variation seen in the dataset (most of the information related to the dataset). PCA is a linear feature extraction tool that helps to maps data to a low dimensional space. This is done by linearly transforming correlated variables into a smaller number of uncorrelated variables, which is performed by projecting the original data into a reduced space using eigenvectors of the correlation matrix (Principal Components). This algorithm is widely used for feature selection when it is highly desired to reduce the number of features (curse of dimensionality) while maintaining a high degree of information on the dataset. Another use of this technique is, as mentioned before, for data visualization.

**Frequent Item:**

This is also called Association rule. This algorithm is characterized by a series of techniques aimed at uncovering hidden sequential relationships between objects. This by calculating the probabilities of certain turns of events over the other. Some of the main areas where this Unsupervised Learning Technique is used are: support measure to show how popular an item is by the proportion of transaction in which it appears, also to suggest which items should be bundled together (purchasing patterns), and last but not least to fuel recommendation engines.

**Singular Value Decomposition:**

SVD is a dimensionality reduction algorithm used for exploratory and interpretability purposes. It is an algorithm that highlights the significant features of the information in the dataset and puts them front and center for further operation. In linear algebra, the Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices (its constituent parts). In a way, SVD is appropriating relevant elements of information to fit a specific cause. SVD is used to extract certain types of information from a dataset, make suggestions for a particular user in a recommender engine system or to curate ad inventory for a specific audience.

1. **Construct a simple X-Y data set from this data set using X = Age and Y = BMI**

Make number of rows = 20.

Explain how you would construct a dendrogram using hierarchical clustering.

Data for this example:

|  |  |  |
| --- | --- | --- |
| Number | Age | BMI |
| 1 | 19 | 27.9 |
| 2 | 18 | 33.77 |
| 3 | 28 | 33 |
| 4 | 33 | 22.705 |
| 5 | 32 | 28.88 |
| 6 | 31 | 25.74 |
| 7 | 46 | 33.44 |
| 8 | 37 | 27.74 |
| 9 | 37 | 29.83 |
| 10 | 60 | 25.84 |
| 11 | 25 | 26.22 |
| 12 | 62 | 26.29 |
| 13 | 23 | 34.4 |
| 14 | 56 | 39.82 |
| 15 | 27 | 42.13 |
| 16 | 19 | 24.6 |
| 17 | 52 | 30.78 |
| 18 | 23 | 23.845 |
| 19 | 56 | 40.3 |
| 20 | 30 | 35.3 |

Chart, scatter chart

Description automatically generated

**Fig.1 BMI Vs Age**

The construction of a dendrogram can be done using 2 methods: Agglomerative clustering & Divisive Clustering. In the first one we start by considering every individual point as a subcluster and after that we define a distance/similarity metric to measure the distance between all the pairs of data of subclusters and at each iteration we merch the closest 2 subclusters. We stop until all the data belongs to the same cluster. The second approach starts by considering the system as a one cluster and then continue splitting the data into smaller clusters until the final size of the of each cluster is each data point.

In the case of Divisive Clustering basically we will perform k-means on each intermediate cluster until the desired minimum number of clusters is reach (for a small dataset as this one this can be k = n). While we are doing this splitting of the data into different clusters based on the inter-distance between the point and the cluster (as per k-means) we must pay attention to the change in the y distance as the number of clusters is getting bigger and bigger.

In the case of Agglomerative clustering, we could start by making a proximity matrix, which will tell us the distance (based on the selected similarity metric) between the points. From this we will look at the smallest distance and then update the proximity matrix and we merge the noted subclusters and accordingly modify the proximity matrix. We repeat the previous step until a single cluster is left.

Final Dendrogram for this dataset:

Chart, histogram, box and whisker chart

Description automatically generated

**Fig.2 Dendrogram for noted dataset (x axis is related to datapoints and y axis is related to the distance between the points)**

See <https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8>