3. Clustering concepts

Distance metric

Distance Metrics

Measuring similarity or distances between different data points is fundamental to many machine learning algorithms

- unsupervised learning problems (i.e. K-means method in clustering)
- supervised learning methods (i.e. K-Nearest-Neighbor) and

- Distance measures are functions that define a distance , between any two data instances and for measuring how similar the instances are.

Distance Metrics...

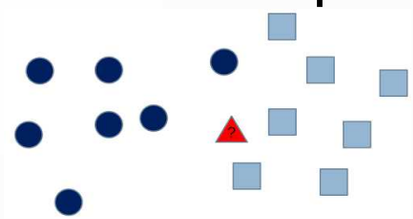
- Distance measures satisfy the following three properties:
  - For any instance , distance with itself is zero, $d(x_i, x_i) = 0$
  - For an instance pairs and , the distance is non-negative and symmetric, $d(x_i, x_j) \geq 0$ and $d(x_i, x_j) = d(x_j, x_i)$
  - Distance measure follows triangular inequality

$$d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k)$$

- Distance measures satisfying above properties are also known as Distance Metrics

Distance Metrics...

- Example 1: Nearest Neighbour Classification
  - Using distance to find the label of the new data point (the red triangle - Square or circle?)



- Example 2: Image retrieval
  - Animal types in NUS Wide Animal dataset

  - given a new image like the image of a cat, can we fetch all cat images from the dataset? (yes! with the help of distance measurements)

Distance Measurement Types

- Euclidean distance
  - ordinary straight-line distance between two points in Euclidean space
  - For any two data instances, represented by d-dimensional feature vectors x_i, x_j, their Euclidean distance is

$$d_{Euclidean}(x_i, x_j) = \left((x_{i,1} - x_{j,1})^2 + \ldots + (x_{i,D} - x_{j,D})^2\right)^{1/2}$$

  - For example, consider these two vectors:

$$x_1 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix} \text{ and } x_1 = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 0 \\ 2 \end{bmatrix}$$

$$d(x_1, x_2) = \left((1-0)^2 + (1-2)^2 + (2-2)^2 + (1-0)^2 + (0-2)^2\right)^{\frac{1}{2}}$$

$$= \sqrt{(1+1+0+1+4)} = \sqrt{7} = 2.65 \text{ (approx)}$$

Distance Measurement Types

- Other distances
  - Cosine distance
  - Mahalanobis distance
  - Cityblock/Manhattan distance
  - Minkowski distance

Clustering concepts

Clustering of Data

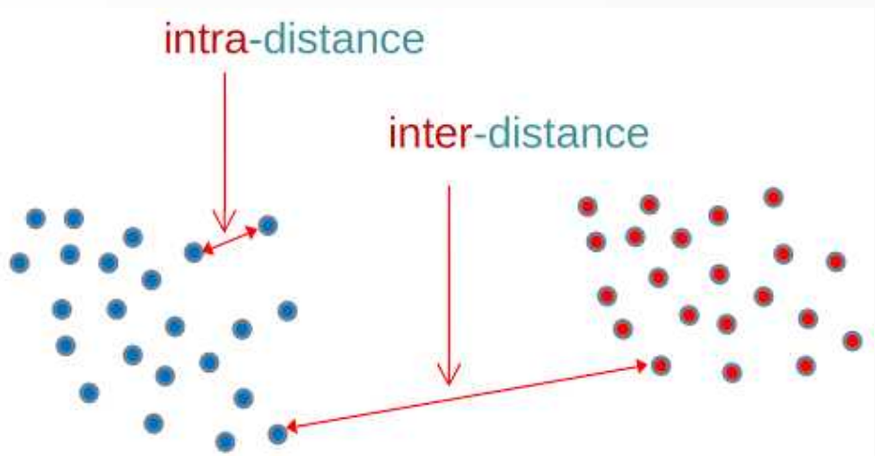
Clustering and It's Applications

- Humans are encoded to see patterns in everything. (related to ML ?!)
- Did I just saw a huge puffy white duck in the sky?

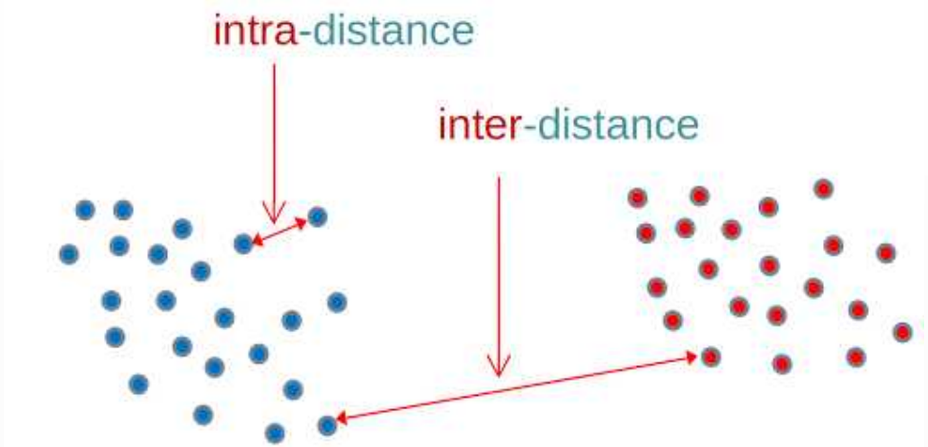
- You are probably right! Our brain prefers patterns and we always look for patterns!
- It looks that our brains do clustering unconsciously

Clustering Algorithms

- How do we teach a computer to do this?
- Goal of clustering algorithms are:
  - Group objects of similar properties together
  - Discover interesting clusters and groups in the data
  - Find valid organisation of the data
- In other words, we can define two algorithmic goals:
  - Minimise intra-distance (distance between points in the same cluster)
  - Maximise inter-distance (distance between points from different clusters)



Clustering Algorithms...



- Now we can define a generic set-up based on our current understanding from clustering methods:
  - Step 1: define a distance metric between objects
  - Step 2: define an objective function that gets us to our clustering goal
  - Step 3: devise an algorithm to optimise the objective function

How Kmeans Works

- The most popular clustering algorithm; simple and fast
- was independently discovered in 60s and 70s by Steinhaus (1955), Lloyd (1957), Ball and Hall (1965) and McQueen (1967)
- Kmeans
  - stores k centroids
  - A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.
  - KMeans searches for the best centroids by alternating between two methods:
    - Assigning data points to clusters based on the current defined centroids (points which are the centre of a cluster).
    - Choosing centroids based on the current assignment of data points to clusters

Limitations of Kmeans

- Most important limitations of Simple Kmeans are:
  - Random initialisation means that you may get different clusters each time. As a solution, we can use Kmeans++ initialisation algorithm to initialise it better.
  - We have to supply the number of clusters beforehand. We can use Elbow method to choose K, but it may not be straightforward.
  - It cannot find clusters of arbitrary shapes.
  - It cannot detect noisy data points, i.e. they should not be taken into account for cluster analysis. K-median is less affected but cannot identify them.

Kmeans with Kmeans++

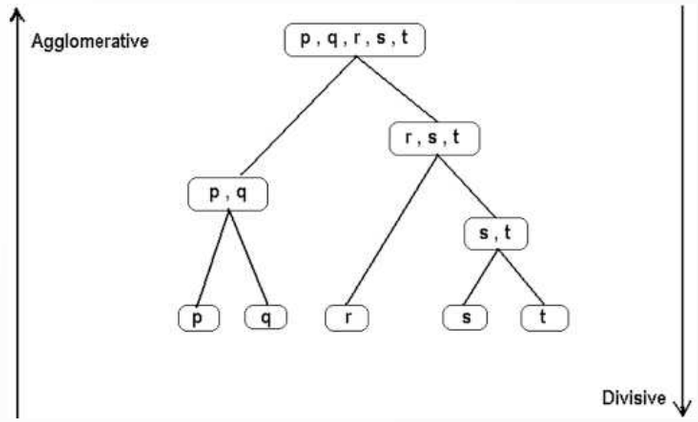
- Kmeans++ is an algorithm for choosing the initial cluster's centre values or centroids for the Kmeans clustering algorithm
  - K-means++ starts with allocating one cluster centre randomly and then searches for other centres given the first one
  - Choose one centroid uniformly at random from dataset
  - Let be the shortest distance from a data point to the closest centroid we have already chose
  - Choose a new centroid from the dataset with probability of $\frac{D^2(x_i)}{\sum_i D^2(x_i)}$
  - Now repeat previous step until we have initialised centroids

Other Clustering Algorithms...

- Hierarchical Clustering:
  - clusters that have a predetermined ordering
  - Two types -
    - **Agglomerative Clustering (Bottom-up)**
      - each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy
    - **Divisive Clustering (Top-down)**
      - all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy

Other Clustering Algorithms...

- Agglomerative Clustering:
  - At the bottom of the tree, at the starting point each of the characters p,q,r,s,t are assigned into a single separate cluster
  - As we go up to the higher levels, the closest characters are formed another cluster. i.e. s,t and p,q.
  - At the next level we can notice r,s,t are making a cluster
  - And finally at the top of the tree, all the characters are in one single cluster p,q,r,s,t
  - How to find the closest cluster pairs? i.e. how to find the distance between two sub-clusters in the middle of the tree?



Other Clustering Algorithms...

- Agglomerative Clustering: four ways to find distance -
  - Single-link: It is a distance between closest points
  - Complete-link: Distance between the furthest points
  - Centroid: Distance between the Centroids
  - Average-link: Average distance between pairs of elements from across cluster pairs

Other Clustering Algorithms...

- Divisive Clustering
  - Same as the Agglomerative Clustering in this type of clustering, initially all data instances are put in the same cluster
  - For splitting, we can use any clustering algorithm that produces at least two clusters (e.g. Kmeans) to find 2 clusters
  - The process is continued until each data instance is separate