f506 Study Guide

Lecture 1 Introduction:

- "All models are wrong, but some are useful"
  - Not every example of a problem fits all its specifications, but it can be useful in identifying some information
- Positive examples: examples that follow a hypothesis model
- Negative examples: examples that do not follow a hypothesis model
- Models are a function of features we've extracted from the data set
- Data science workflow
  - Process data -> explore data -> extract features -> create model
- Types of data:
  - M-dimensional points/vectors
    - 3 tuples of data ex: (name, age, income)
  - Graphs
    - Nodes connected by edges
    - Can be represented in an adjacency matrix/ list
  - Images
    - Grids of pixels
  - Text
    - List of words
  - Corpus of documents
    - Lots of documents described through a table
- Types of learning
  - Unsupervised Learning
    - Find interesting structure within the data (clustering)
    - Goals:
      - better understand/ describe data
        - Find anomalies
        - Data exploration/visualization
      - Extract features
      - Fill in gaps in data
      - Make algorithms faster
        - Get rid of noise
  - Supervised learning
    - Making predictions based on known inputs and outputs
    - Making predictions on new/unknown data based upon old data

Lecture 2 – distance and similarity

- Data points are rows
- Features are columns
- Feature space
  - All possible values for the collection of features in our data set
  - Feature space is in the Euclidian plane defined by vectors

- Dissimilarity
  - Method of comparing data points to see how unalike they are
  - Dissimilarity function: takes two objects and returns a large value if these objects are dissimilar
  - A special type of Dissimilarity function is a distance function
- Distance function:
  - D is a distance function only if
    - D(I,j) = 0   I and j are the same
    - D(I,j) = D(j,i)
    - D(I,j)<= d(I,k) +d(k,j)   where k is some middle point

- Minkowski Distance
  - $L_p(x, y) = (\sum_{i=1}^{d} |x_i - y_i|^p)^{\frac{1}{p}}$
  - P= 2 is the Euclidian distance
  - P = 1 is the Manhattan distance
  - d=dimensions (characteristics/attributes)

- Jaccard Similarity
  - $JSim(x, y) = \frac{|x \cap y|}{|x \cup y|}$

- Jaccard Distance
  - $JDist(x, y) = 1 - \frac{|x \cap y|}{|x \cup y|}$
- Similarity Function
  - Similarity function : is a function that takes two objects(data points) and returns a large value if these objects are similar
  - $s(x, y0 = \cos(\theta)$ where theta is the angel between x and y
  - two proportional vectors have a cosine similarity of 1
  - two orthogonal vectors have a cosine similarity of 0
  - two opposite vectors have a cosine similarity of -1
- when should you use cosine (dis)similarity over Euclidean distance?
  - When direction matters more then magnitude
- Norms
  - Norm(x) = distance(0,x) or d(0,x) = $\|x\|$ (they mean the same thing)
  - Distance between a and b : d(a,b) = $\|a - b\|$

Lecture 4 Clustering – Kmeans

- Clustering: a grouping or assignment of objects (data points) such that objects in the same group/cluster are similar to each other or dissimilar to objects in other groups
    - Applications
        - Outlier detection /anomaly detection
            - Data cleaning/ processing
        - Feature extraction
        - Filling in gaps in data
- Types of clustering
    - Partitional
        - Each object belongs to exactly one cluster
        - Goal: partition dataset into k partitions
    - Hierarchical
        - A set of nested clusters organized in a tree
    - Density based
        - Defined based on the location of density points
    - Soft Clustering
        - Each point is assigned to every cluster with a certain probability
- Cost function
    - Way to evaluate and compare solutions

$$\sum_{i}^{k} \sum_{x \in C_i} d(x, \mu_i)^2$$

- K-Means
    - Given a dataset X= {x1… xn}, d the Euclidian distance, and k , find k centers {μ1 ..μk} that minimize the cost function
    - This is an NP-hard problem

- K-means – Lloyd's Algorithm
    - Randomly pick k centers
    - Assign each point in the dataset to its closest center
    - Compute the new centers as the means of each cluster
    - Repeat 2 and 3 until convergence
    - !!!! Lloyds algorithm always converges !!!! but not always to the optimal solution

Lecture 5 Kmeans++

- K-means++
    - Start with a random center
    - Let D(x) be the distance between x and the closest of the centers picked so far. Choose the next center with probability proportional to $D(x)^2$
    - How to choose the right k?
        - Iterate through different values of k (elbow method)
            - The graph is y = cost and x=k

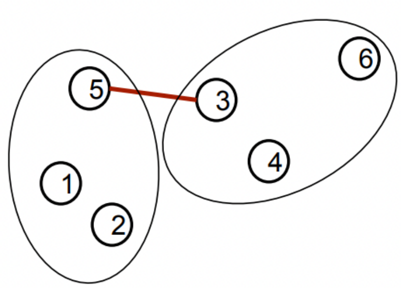
            - The elbow in the graph is the point of diminishing returns
        - Use empiricle / domain specific knowledge
        - Metric for evaluating clustering output
    - Goal
        - Find a clustering so that similar points are in the same cluster and dissimilar points are in different clusters
    - Silhouette Scores
        - For each data point i,
            - ai is the mean distance from point I to every other point in its cluster
            - bi is the smallest mean distance from point I to every point in another cluster
        - the overall Silhouette Score is:
            - $s_i = (b_i - a_i)/max(a_i, b_i)$

Lecture 6 – Hierarchical Clustering

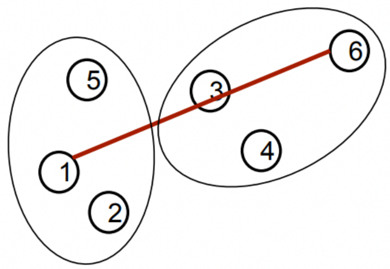
- Two types of hierarchical clustering
    - Agglomerative
        - Start with every point in its own cluster
        - Compute the distance between all pairs of clusters
        - Merge the two closest clusters
        - Repeat 3 and 4 until all points are in the same cluster
    - Divisive
        - Start with every point in the same cluster
        - At each step split until every point is in its own cluster
- Distance functions
    - Lets define
        - Distance between points: d(p1, p2)
        - Distance between clusters: D(C1, C2)
    - Single Link Distance
        - Single link distance is the minimum of all the pairwise distances between a point from one cluster and a point from the other

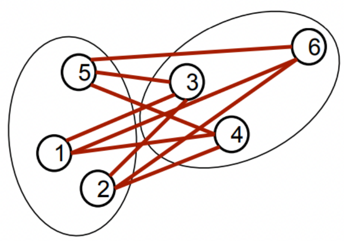
$$D_{SL}(C_1, C_2) = \min \{d(p_1, p_2) \mid p_1 \in C_1, p_2 \in C_2\}$$

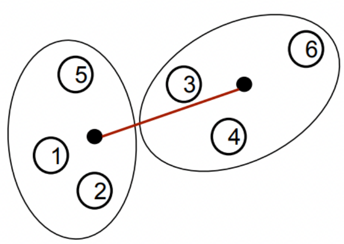
        -
        - Dependent on the choice of d (dimensions)



        -

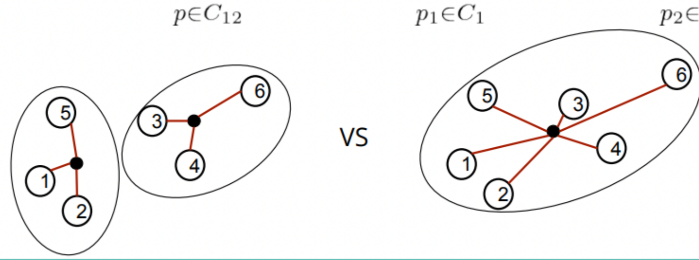
- Complete- Link Distance
  - Is the maximum of all the pairwise distances between a point from one cluster and a point from the other cluster
  - $$D_{CL}(C_1, C_2) = \max \{d(p_1, p_2) \mid p_1 \in C_1, p_2 \in C_2\}$$
  - 

- Average Link Distance
  - Is the average of all the pairwise distances between a point from one cluster and a point from the other cluster
  $$D_{AL}(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{p_1 \in C_1, p_2 \in C_2} d(p_1, p_2)$$
  - 

  - Less susceptible to noise and outliers but is more biased toward globular clusters
- Centroid Distance
  - The distance between the centroids of the clusters
  $$D_C(C_1, C_2) = d(\mu_1, \mu_2)$$
  - 

- Ward's Distance
  - The difference between the spread / variance of the points in the merged clusters and the unmerged clusters

$$D_{WD}(C_1, C_2) = \sum_{p \in C_{12}} d(p, \mu_{12}) - \sum_{p_1 \in C_1} d(p_1, \mu_1) - \sum_{p_2 \in C_2} d(p_2, \mu_2)$$

- 

Lecture 7 – Density- Based Clustering

- Goal: cluster points that are densely packed together
- Density: Given a fixed radius **ε** around a point, if there are at least min_pts number of points in that area, then this area is dense.
- Core points: if its **ε** neighborhood contains at least min_pts
- Border point: if its in the **ε** neighborhood of a core points
- Noise points: if it is neither a core nor border point
- DBScan Algorithm:
  - 1. Find the **ε**-neighborhood of each point
  - 2. Label the point as core if it contains at least min_pts
  - 3. For each core point, assign to the same cluster all core points in its neighborhood (crux of the algorithm)
  - 4. Label points in its neighborhood that are not core as border
  - 5. Label points as noise if they are neither core nor border
  - 6. Assign border points to nearby clusters