Homework 2

Charles Richardson (UFID : 73112398)

Syed Faizan Ali (UFID : 26592828)

Q1: Market basket transactions

1. Items: Bread, Butter, Milk, Beer, Coke, Apple | Item count = d = 6
    - Therefore, maximum number of candidate association rules is 2^d, or 64 association rules
2. Maximal itemsets with minsup threshold >= 2
    - {Bread, Butter}
    - {Bread, Milk}
    - {Bread, Beer}
    - {Bread, Apple}
    - {Milk, Beer}
    - {Beer, Apple}
    - {Milk, Beer, Apple}
3. Closed Frequent Itemsets
    - Identical as listed above
4. Support and confidence
    - Support({Bread}) = 4/5
    - Support({Milk}) = 3/5
    - Support({Bread, Milk}) = 2/5
    - Confidence rule of {Bread} -> {Milk} = 1/2
    - Confidence rule of {Milk} -> {Bread} = 2/3

Q2: Apriori Property

The Apriori Principle states that if an itemset is frequent, all of its subsets will be frequent. We can use subsets instead of complete itemsets if doing so reduces the size and complexity of the computation.

Q3: Cluster Methods on point sets

1. DBSCAN

a. DBSCAN would struggle to detect the low density patterns of the face. It does well with high densities, but the patterns on the face would lost likely be detected as noise.

b. DBSCAN could successfully find the clusers corresponding to the patterns represented by the nose, eyes, and mouth separately because of its ability to differenciate densities. The high density patterns will stick out significantly to DBSCAN.

2. K-means

a. K-means would have a hard time extracting these patterns from the face. If the initial cluster points were made directly on top of the pattern centeriods, it may get close, but it would likely not be very high quality.

b. K-means would do much better in this plot since the points in the pattern are much denser than the rest of the face. This would be a straightforward cluster.

Q4: Clustering short documents

1. I would use the Cosine Similarity method, as the feature vectors are numeric, and the words in the documnents are generally symmetric, which cosine is better for.

2. DB-Scan could be used with cosine similarity to detect document clusers, but there would need to be a firm limit on the dimensionality, as DB-Scan does not do well with high dimensional data.

3. Redesigning the "computing centeriod" step would consist of choosing the point that has the highest cosine similarity with all the points in the cluster, finding the mean of the normalized data points. For "assigning samples to closest centeroid", we simply assign the centeriod as one data point and the cluster point as the other and compute cosine similarity with that.

Q5: Finding output anomolies

1. K = 2. The point with the radius drawn around it and the 3 points on the right would be classified as outliers, since they only have 2 neighbors each and are far away from the other cluster of points.

2. K = 1. Only the example point would be classified as an outlier, since it is far away from the other cluster of points and has no neighbors. The other 3 points would be classified as inliers since they have 2 neighbors.

3. The effect of the parameter K heavily affects the outlier detection in a sample with sparse data. The higher the value of K, the more points will be classified as outliers, since the number of neighbors is a factor in the classification.

4. Only the point with the radius drawn around it would be classified as an outlier, since it is furthest away from the other cluster of points and has no neighbors and K = 1

5. The point with the radius drawn around it and the point on the furthest right would be classified as outliers, since they are far away from the other cluster of points and have no neighbors and K = 2

Q6: Challenges with outlier detection with supervised classification method

Data is noisy, and the outliers are similar to the data. This could be addressed by using a very large decision tree; or an NN, which could learn the patterns in the data better.

Q7: Prove briefly in math that each step in K-Means algorithm (assigning point to closest centroid, updating centroid with mean of points within clusters) is a greedy choice to minimize the object function (sum of squared distances).

Let $X = x1, x2, \ldots, xn$ be the set of n data points we want to cluster.

Let K be the number of clusters we want to find.

We want to minimize the sum of squared distances between each point and its closest centroid.

$J = \sum_{i=1}^{n} a_i ||x_i - \mu c_i||^2$

Assign each data point to the closest centroid. Let c_i be the index of the centroid closest to x_i.

Then, $c_i = argmin_j ||x_i - \mu_j||^2$

Minimizes the sum of squared distances between each point and its closest centroid. **This is a greedy choice to minimize J.**

Update the centroids with the mean of the points within the cluster. Let μ_j be the mean of the points in cluster j.

Then, $\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$

This step minimizes J because it updates the centroids to be the mean of the points in the cluster. **A greedy choice to minimize J.**

Specifically,

$J = \sum_{i=1}^{n} ||x_i - \mu c_i||^2$

$$= \sum_{k=1}^{K} \sum_{i \in Sk} ||x_i - \mu c_i||^2$$

$$= \sum_{k=1}^{K} \sum_{i \in Sk} ||x_i - \mu_k + \mu_k - \mu c_i||^2$$

$$= \sum_{k=1}^{K} \sum_{i \in Sk} ||x_i - \mu_k||^2 + \sum_{k=1}^{K} \sum_{i \in Sk} ||\mu_k - \mu c_i||^2$$

where we have used the fact that uc_i is the centeroid of the k-th cluster Sk and we expanded the square term using binomial expansion. **This is the final greedy choice to minimize J.**

References

- https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a
- https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68
- Zhe Jhang Class Notes