CSCI-B 565: Assignment 4

Abhinandan Sampathkumar

Problem 1:

The minimum points to plot a ROC would be one. Since we get one point we can use that as a reference to plot our graph. It is true that the graph might look different with more classification points, but with one point we can plot a ROC graph and calculate the area under it. The maximum points can be N, we can take all classification points to get more accurate curve.

Problem 2:

The problem requires us to merge clusters which are similar, using a new centroid, which has to be computed using the distances between points.

We need to find the new centroid to merge two clusters which will be the centroid of the new merge cluster. For this we will have to calculate the new centroid, say Ci. Let's Xi and Yi are points of two different clusters. We need to find the value for Ci such that it minimizes the intra cluster distance.

Using SSE and considering the minimum distance between the points in X and Y clusters. We can compute Ci using Xi and Yi, which are values of clusters X and Y. With this new centroid we can merge clusters.

Problem 3: Apriori Algorithm

a) Fk−1 × F1 and Fk−1 × Fk−1

I have implemented both Fk−1 × F1 and Fk−1 × Fk−1 methods for candidate generation. The user is asked for an input to choose one of the methods that has to be used for candidate generation and the candidate generation will be done accordingly.

In my code I print the candidates generated and the frequent itemsets and the total number of itemsets considered for Rule generation.

For a support of 0.15 on car eval data I get the below output.

```
enter 1 for Fk-1*Fk-1 and 2 for Fk-1*F1 1
enter 1 for confidence and 2 for lift  1
Filepath is:car.txt
No of transcations  1728
No of columns 15
*********candidateList count is***** 15
*********candidateList is********** {0: 'vhigh', 1: '2', 2: 'small', 3: 'low', 4: 'unacc', 5: 'med', 6: 'high', 7: 'big', 8: '4
******itemset count is********* 13
******itemset is************ {'acc': 384.0, 'med': 1296.0, '5more': 432.0, 'big': 576.0, 'vhigh': 756.0, 'high': 1080.0, '3': 43
*****candidateList count is******** 78
*****candidateList is************** set([frozenset(['big', '2']), frozenset(['high', 'more']), frozenset(['big', 'low']), frozen
****itemset count is****** 47
****itemset is************ set([frozenset(['big', '2']), frozenset(['high', 'more']), frozenset(['big', 'low']), frozenset(['high
*****candidateList count is******** 194
*****candidateList is************** set([frozenset(['small', 'unacc', 'med']), frozenset(['vhigh', '2', '4']), frozenset(['acc',
****itemset count is****** 25
****itemset is************ set([frozenset(['small', 'unacc', 'med']), frozenset(['small', 'unacc', 'low']), frozenset(['small',
*****candidateList count is******** 38
*****candidateList is************** set([frozenset(['high', '4', 'small', 'unacc']), frozenset(['high', '4', 'unacc', 'vhigh']),
****itemset count is****** 2
****itemset is************ set([frozenset(['med', '2', 'low', 'unacc']), frozenset(['high', 'med', '2', 'unacc'])])
*****candidateList count is******** 1
*****candidateList is************** set([frozenset(['med', 'high', '2', 'low', 'unacc'])])
****itemset count is****** 0
****itemset is************ set([])
*******total itemset count is***** 87
```

b) 3 Datasets, Comparison and Discussion

The 3 datasets that I have used are Car Evaluation, Nursery and Contraception Datasets. Nursery has records over 10000. The other two have records over 1000.

Below are tables for Number of candidates generated using both the methods on 3 datasets for different values of minimum support.

Minimum Support (Threshold value)	Car Fk-1*Fk-1 | Fk-1*F1
0.05	1115 | 1491
0.15	326 | 528
0.25	99 | 173

Minimum Support (Threshold value)	Contraception Fk-1*Fk-1 | Fk-1*F1
0.01	16565 | 20852
0.05	153 | 213
0.08	89 | 105

Minimum Support (Threshold value)	Nursery Fk-1*Fk-1 | Fk-1*F1
0.05	4832 | 7054
0.15	711 | 1205
0.25	121 | 153

In Fk-1*Fk-1 method the candidates are less in comparison with Fk-1*F1 because in Fk-1*Fk-1 the candidates are generated by checking with k-1 frequent lists, so in every round of candidate generation we will generate combinations using previous round frequent itemsets. Where as in Fk-

1*F1, the frequent list and F1 lists are used to generate next round of lists. Since F1 has of the items in its list, Fk-1*F1 always generated more candidates.

c) Frequent Closed Itemsets and Frequent Maximal Itemsets

For Car DataSet

Minimum Support (Threshold value)	Frequent Itemsets	Frequent Closed Itemsets	Maximal Itemsets
0.05	1115	224	118
0.15	326	45	40
0.25	32	17	10

For Contraception Dataset

Minimum Support (Threshold value)	Frequent Itemsets	Frequent Closed Itemsets	Maximal Itemsets
0.01	1033	795	34
0.05	61	54	4
0.08	35	33	1

For Nursery Dataset

Minimum Support (Threshold value)	Frequent Itemsets	Frequent Closed Itemsets	Maximal Itemsets
0.05	605	239	118
0.15	79	22	4
0.25	17	10	3

As you can observe, all maximal are closed and all closed are frequent.

d) 3 levels of support and confidence and comparison with brute force

For Car Eval Dataset

Min Support and Min Confidence	No of rules for Min Support and Min Confidence	In Brute Force
0.1and 0.5	267	752
0.15 and 0.6	82	272
0.2 and 0.8	6	146

For Contraception Dataset

Min Support and Min Confidence	No of rules for Min Support and Min Confidence	In Brute Force
0.1 and 0.5	pp	180
0.05 and 0.7	180	366
0.2 and 0.8	46	180

For Nursery Dataset

Min Support and Min Confidence	No of rules for Min Support and Min Confidence	In Brute Force
0.15 and 0.6	19	106
0.12 and 0.4	91	186
0.05 and 0.8	395	2466

As threshold of support and confidence increases the no of rules will be reduced as we got good valid association rules at higher confidence. In brute force method, since we do not do pruning based on confidence we get more rules.

e) Top ten association rule for a support and different confidence.

For Car Eval dataset with Support threshold value 0.15 we get below top 10 rules for different values of confidence. I got around 140 rules for confidence value 0.6.

```
('2',) ==============> ('med',) ,with confidence, 0.750
('4',) ==============> ('med',) ,with confidence, 0.750
('med', 'vhigh') ==============> ('unacc',) ,with confidence, 0.779
('small',) ==============> ('unacc',) ,with confidence, 0.781
('vhigh',) ==============> ('unacc',) ,with confidence, 0.810
('high', '2') ==============> ('unacc',) ,with confidence, 0.856
('med', '2') ==============> ('unacc',) ,with confidence, 0.864
('2',) ==============> ('unacc',) ,with confidence, 0.877
('2', 'low') ==============> ('unacc',) ,with confidence, 0.887
('2', 'vhigh') ==============> ('unacc',) ,with confidence, 0.926
total rules 55
```

Below are the top ten rules for Contraception data with support 0.15 and confidence 0.6. I got over 140 rules and these are the top ten rules.

```
('0', '2') ==============> ('1',) ,with confidence, 0.977
('4',) ==============> ('1',) ,with confidence, 0.977
('2',) ==============> ('1',) ,with confidence, 0.978
('0', '3', '4') ==============> ('1',) ,with confidence, 0.978
('3', '4') ==============> ('1',) ,with confidence, 0.979
('0', '3', '2') ==============> ('1',) ,with confidence, 0.980
('3', '2') ==============> ('1',) ,with confidence, 0.981
('0',) ==============> ('1',) ,with confidence, 0.981
('0', '3') ==============> ('1',) ,with confidence, 0.984
('3',) ==============> ('1',) ,with confidence, 0.984
total rules 140
**********CLOSED ITEMSETS**********
```

Below are the association rules for nursery data for a support count of 0.25.

```
'2',) ==============> ('convenient',) ,with confidence, 0.667
'improper',) ==============> ('convenient',) ,with confidence, 0.667
'completed',) ==============> ('convenient',) ,with confidence, 0.667
'complete',) ==============> ('convenient',) ,with confidence, 0.667
'priority', 'slightly_prob') ==============> ('convenient',) ,with confidence, 0.669
'priority', 'nonprob') ==============> ('convenient',) ,with confidence, 0.669
'priority', 'pretentious') ==============> ('convenient',) ,with confidence, 0.669
'priority',) ==============> ('convenient',) ,with confidence, 0.671
'priority', 'problematic') ==============> ('convenient',) ,with confidence, 0.677
'priority', 'recommended') ==============> ('convenient',) ,with confidence, 0.680
'priority', 'great_pret') ==============> ('convenient',) ,with confidence, 0.686
```

Even for different confidence threshold values we get same top ten Rules as the support count will be same. Although I ran for different confidence values I have reported rules for Top 10 which is same at different confidence levels.

'supportThreshold' and 'confidenceThreshold' are the variable names for support and confidence in my code. You can change these values to compare different results and rules

The results of the rules depend on the dataset and their occurrence in the dataset. Since these are not exactly association dataset and we use classification and regression dataset, we get some peculiar rules and if we find few elements occurring in multiple examples, it is very likely that we find a rule between them. As the support and confidence threshold increases the quality of association rules also is improved.

f) Top 10 rules using lift

In my program the user will be asked to input if he wants to generate rules based on confidence or based on lift

Rules generated with Car dataset using Lift

```
('2', 'vhigh') ==============> ('unacc',) ,with confidence, 1.322
('unacc',) ==============> ('vhigh', '2', 'low') ,with confidence, 1.327
('vhigh', '2', 'low') ==============> ('unacc',) ,with confidence, 1.327
('unacc',) ==============> ('small', '2') ,with confidence, 1.339
('small', '2') ==============> ('unacc',) ,with confidence, 1.339
('2',) ==============> ('high', 'med', 'unacc') ,with confidence, 1.357
('high', 'med', 'unacc') ==============> ('2',) ,with confidence, 1.357
('acc',) ==============> ('med', '4') ,with confidence, 1.417
('med', '4') ==============> ('acc',) ,with confidence, 1.417
('acc',) ==============> ('more',) ,with confidence, 1.453
('more',) ==============> ('acc',) ,with confidence, 1.453
('acc',) ==============> ('high', '4') ,with confidence, 1.508
('high', '4') ==============> ('acc',) ,with confidence, 1.508
```

Rules generated with Contraception Data using Lift

```
('0',) ==============> ('3', '4') ,with confidence, 1.023
('3', '4') ==============> ('0',) ,with confidence, 1.023
('0',) ==============> ('3', '2', '4') ,with confidence, 1.023
('3', '2', '4') ==============> ('0',) ,with confidence, 1.023
('0',) ==============> ('1', '4') ,with confidence, 1.024
('1', '4') ==============> ('0',) ,with confidence, 1.024
('0',) ==============> ('4',) ,with confidence, 1.025
('4',) ==============> ('0',) ,with confidence, 1.025
('0',) ==============> ('1', '2', '4') ,with confidence, 1.026
('1', '2', '4') ==============> ('0',) ,with confidence, 1.026
('0',) ==============> ('2', '4') ,with confidence, 1.026
('2', '4') ==============> ('0',) ,with confidence, 1.026
```

Rules generated with Nursery Data using Lift

```
('convenient', 'critical') ==============> ('spec_prior',) ,with confidence, 1.257
('priority', 'inconv') ==============> ('spec_prior',) ,with confidence, 1.259
('spec_prior',) ==============> ('priority', 'inconv') ,with confidence, 1.259
('spec_prior',) ==============> ('critical',) ,with confidence, 1.259
('critical',) ==============> ('spec_prior',) ,with confidence, 1.259
('spec_prior',) ==============> ('priority', 'critical') ,with confidence, 1.485
('priority', 'critical') ==============> ('spec_prior',) ,with confidence, 1.485
('great_pret',) ==============> ('spec_prior',) ,with confidence, 1.500
('spec_prior',) ==============> ('great_pret',) ,with confidence, 1.500
('very_crit',) ==============> ('spec_prior',) ,with confidence, 1.877
('spec_prior',) ==============> ('very_crit',) ,with confidence, 1.877
```

The issue with confidence is that it does not consider the support in the rule consequent. The solution to this is by measuring lift. Lift addresses this by considering the consequent of the rule.

Let's say a rule X, Y -> Z. It consider the support of X, Y only; but with considering the support of Z we will get better rules.

This improves the association rule.

Problem 4

a) An impossibility Theorem for Clustering

Introduction

The paper primarily focuses on the idea that there cannot be one clustering function that satisfies three properties presented by Kleinberg. The author states that there is no single unification method to address this problem. Clustering is a concept of grouping a large set of objects into partition sets, where all the objects in a partition sets are similar. The paper talks about the difficulty to find a unification method using the of impossibility theorem.

An axiomatic framework is used for the clustering. A clustering function, CF, takes n points as input and produces P partitions, which are the clusters. This clustering function, CF, should satisfy three properties - scale-invariance, richness and consistency.

The impossibility theorem is composed of these three properties.

Scale-Invariance states that the clustering function should not depend on the change in units of distance between points, i.e., it should not vary even if the distance between the points are scaled by a constant value.
Richness states that an ideal clustering function should be able to produce partitions even when the distance between the points are not given in the set.
Consistency states the if we reduce the distance between points in the same cluster and increase the distance between the clusters, the clustering logic should not change, meaning the function has to be consistent.

Theorem 2.1 states that for any set with elements k, where k>=2, there is no clustering function which satisfies all three properties. He uses the single-linkage procedure to discuss the theorem. Single-linkage works by building a connected graph betweenuntil a stopping condition is reached. At this point the graph is returned, which is a partition

There are 3 types of stopping condition for this:

- K-cluster stopping condition: Stop building the graph after k nodes.
- Distance-r stopping condition: Edges of weight r.
- Scale-α stopping condition: Do not add edges more than αp*, where p* is the maximum distance between two points.

These stopping conditions does not satisfy all three properties. K-cluster stopping condition violates richness, Distance-r stopping condition is not Scale-Invariant and Scale-α stopping condition would not allow to maintain consistency. So that proves the impossibility theorem.

He uses these conditions to state theorem 2.2

For a value of k>=1 and k<=n, cluster function qualifies the properties scale-invariance and Consistency with k stopping condition

For a value of r<0 and n>=2, cluster function qualifies the properties richness and Consistency with distance-r stopping

For any positive α value <1, and n >=3, cluster function qualifies the properties scale-invariance and Consistency with scala-α

Antichain of Partitions:

This is stated to further strengthen the impossibility result by using a refined cluster. An antichain is a collection of clusters where each cluster is refined partial order set of another cluster. Theorem 3.1 and 3.2 use this concept of antichain to prove his impossibility result. The author provides substantial mathematical proof for these theorems. 3.1 emphasises on Range(f) being a antichain and 3.2 focuses on every antichains of partition A, which has a clustering function satisfying consistency and scale-invariance, where Range(f) is A.

Centroid-Based Clustering

Clustering based on the centroid of a set is an approach widely used for clustering. This method calculates the centroid for the set of points and goes on building the clusters by grouping the data points based on the centroids. The author shows that this kind of clustering function does not satisfy the consistency property. Centroid based functions like K-median and k-mean too does not satisfy the property, thereby again proving the impossibility theorem.

Relaxing the Properties

Kleinberg talks about the effect of relaxing the 3 properties. He mentions about how relaxing some of the properties allow clustering functions to satisfy those properties which they failed to, before relaxation. As one example, he talks about single-linkage with distance-r stopping function, the clustering function now satisfies a relaxation of scale-invariance. He does provide examples of cases where other properties are also now satisfied with some relaxation.

Opinion

This paper provides an interesting view on clustering. The impossibility theorem poses some doubts, the author presents his three methods which a good clustering function should satisfy. Let's consider the consistency property, with decreasing the distance between points and increasing the distance between clusters, we might reach a point where few clusters are far from all other clusters and they might combine to form one cluster. The consistency property in this case would be ruined as the clusters merged to form another cluster. The richness property is kind of abstract and really not clear. Also the author does not give any real examples while stating his theorems which makes it complicated. The size of the original set and the number of clusters formed also plays an important role while checking the function for satisfying properties. Some desirable changes in one of the properties might allow us to find a clustering function which satisfies all 3 properties. Overall, it's an interesting read and will lead us to read more about clustering and expands our horizon about the concept.

b) Measures of Clustering Quality: A Working set of Axioms for Clustering

Introduction

The paper talks about Impossibility theorem proposed by Kleinberg and opposes the Kleinberg's idea of assessing a clustering function. Rather than clustering functions satisfying some properties, the author states that having a quality measure for the clustering process. The paper introduces Clustering Quality measure and introduces with an analysis of complexity and examples. Authors of this paper oppose on the basis of inherent property being misunderstood and misinterpreted. The paper says that the clustering quality measure will also consider the dataset as well as the clustering.

Detailed Review

The paper introduces the problem in finding a clustering-quality measure, such as k-means short-coming, not being scale invariant which will not help in measuring the clustering quality.
They also mention about the cluster validity from the statistics area and how it does not help in having a clustering quality measure. Since there are no general model for having a measure they come up with a theoretical basis. They will evaluate their measure on both consistency and relevance. Without further explanation on the quality measure they say that they give a computational complexity.

Previous axioms and more axioms…….

After the brief introduction on what the clustering measure will be they continue with definitions and notation that will be used. In addition to satisfying the clustering for the distance function the paper also discusses about satisfying additional requirements. Continuing with the Kleinberg introduction on what the axioms are and what are its implications this paper has a discussion section to bring out the important weakness that is found on the Kleinberg's paper. Consistency is stated to be the fundamental concept behind Kleinberg. After discussing the Kleinberg's theorem and giving an example on finding the weakness.

Authors now bring up the axioms that will be considered for this paper by starting with axioms that are analogous to Kleinberg's axioms. The authors have also introduced a new set of axioms on top of the axioms that are there. The new theorem that they state is helpful for centroid based clustering. Now they introduce Relative Margin which is said to satisfy all the three axioms by the

Kleinberg. Although this satisfies all the three axioms this may have the same flaw as the three included.

Relative Margin

Relative Margin is a Clustering Quality measure which qualifies all the three properties claimed by Kleinberg. Though, the authors haven't provided a proper definition for the measure, they have provided a mathematical proof for the same and claims that the smaller values of relative measure provides better clustering quality.

Soundness and Completeness

The author then defines Soundness and Completeness. Soundness is about consistency, all instances should satisfy each of the axioms. Completeness means that every property is implied by the axioms. The paper then considers this as a challenge stating that there is no clear definition of what a clustering function is. They mention about relaxing soundness and completeness. Also, they argue that the 3 axioms are sound but not complete for class of Cluster Quality Measure. The paper then introduces a new axiom

Isomorphic Invariance

Two clusters are said to be isomorphic if the distance between them is preserving isomorphism. This axiom does not have a corresponding axiom with Kleinberg's axiom. They define Clustering Isomorphism and Isomorphic Invariance. They provide a proof for theorem 3 and also mention that Relative margin quality measure satisfies all of the four axioms.

Examples of Clustering Quality Measures

They introduce two CQM's which satisfy all of the axioms. These measures are related with linkage-based clustering and centre-based clustering.

Weakest Link - The points in the same cluster are often close and strong but the importance is given to the longest link in the cluster. The paper provides two definitions of weakest links. It is a ratio of highest value of longest list over minimum distance between two points.
Additive Margin - This is a quality measure for centre based clustering, whereas weakest link was for linkage-based clustering. It considers the difference for evaluation rather than the ratio. Good clusters get higher values from Additive Margin.

The paper discusses about computational complexity of these measures. All of the measures can be computed in n-polynomial time to measure the quality of a cluster. It discusses about time taken by different measures based on the type of data set used for clustering.

Other Measures

Next, they talk about having variants for measuring quality. It talks having new measures with other features and applying them on sub clusters. If measure m satisfies the axioms, then all variants of m are also said to qualify the axioms, which is great because it lets us come up with many variants of standard measures but will still satisfy all properties.

Now the author talks another type of evaluation which are dependent on the number of clusters. Since the common loss functions do not qualify scale invariance and richness. L-normalization

can be used to satisfy scaling by normalization, as they are dependent on number of clusters. But even after normalisation they do not satisfy richness. They then discuss about Refinement and Coarsening preference, which also won't satisfy richness property. Finally, they state that it is better to use the measures which are independent of cluster, as we would not know the number of clusters always.

In my view, this paper clearly opposes the first one and proves it wrong with very good examples. This paper also provides very good definitions and notations. The axioms are clearly explained too. It talks about how CQM's are better to evaluate clusters but they all have minor issues as well. The paper also does not compare with any other baseline measures, they just tell about how some measures satisfies axioms. Overall, it was good paper which challenges the first interesting paper and also gives new insights about clustering and its measures.