# CSCI B 565: Assignment #4

Due on Saturday, April 9, 2016

*Prof. Predrag*

**Vivek Patani (vpatani)**

## Contents

## Problem 1

Listing 1 shows the pseudocode.

Listing 1: Pseudocode

```
L <- We sort the the input with decreasing scores of f:
Fp <- TP <- 0
R <- 0
i <- 1
while i <= L do
     if f(i) != fprev then
     push (FP/N, TP/P) onto R
     fprev <- f(i)
     end if

     if L[i] is positive:
     TP <- TP + 1
     else FP <- FP + 1
     i = i + 1
end while
push(FP/N,TP/P)
end
```

- ROC can be generated with the help of classifier which predicts class of a data point and the posterior probability.
- The idea is to find the minimum and maximum number of points to find the area under the ROC curve.
- Let us see what the pseudocode says - We first sort the data points on basis of their probability picking each of the distinct probability as the thresholds. This can also sometimes indicate that the maximum number of threshold as n itself as probability of each data point can be taken as threshold.
- The next step we initalise TP & FP(True Positives and False Positives) as zero. As we go on they will be incremented as and when whatever occurs. We then create a Stack R which consists of all the ROC points. The observation also sees that the complexity of our algorithm is $O(n \log n)$.
- Next step is to on until all the points are over. We also need to keep a check on the instances with equal probability.
- When we enter the loop say for example if all the probabilities are equally likely, what happens then? There will be two extremes as either we will only end up having a single point since the if condition will not be TRUE at all.
- Similarly just think that if all the probabilities are unique and unequal then too we will have n values causing us to think of the higher bound.
- To summarise my answer I would say if and only if we have the starting and end point then we need only 1 point to plot the ROC and calculate area. Since I'm assuming we are known to have (0,0) and (1,1) then we can rely on a single point since it completes the curve, but if we have no prior knowledge then we might as well need a minimum of two points both on the opposite axis (For FP and TP). The maximum number of points that we need would be the number of unique posterior probability which may or may not equal to 1.

## Problem 2

Listing 2: Pseudocode

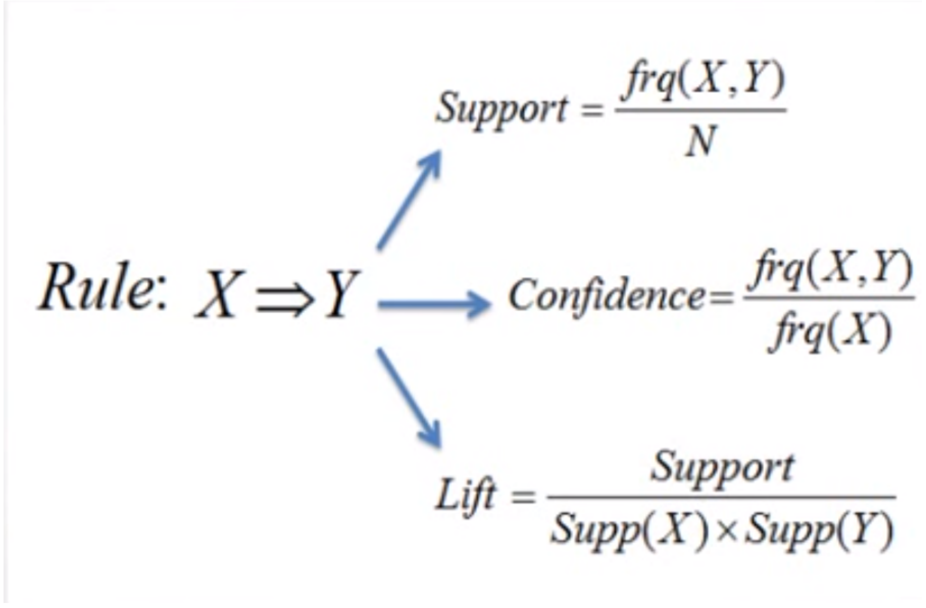
```
n = #number_of_clusters
for each_cluster in n:
     #Calculate SSE for each_cluster
     for every_other_cluster in n:
          if each_cluster != every_other_cluster:
               #Calculate SSE for each_cluster and every_other_cluster
               #find the smallest cluster and add to a list
               #Repeat for all clusters
               #Once all the clusters are done searching process the list
          end if
     end for
end for
Get the smallest pair - cluster for minimum SSE
```

- The idea is to merge m pairs of clusters minimising SSE of all clusters.
- In order to do that we need to start off with the number of clusters, for every cluster go to every other cluster and get the distance, store it.
- Once both the for loops end we will get the list of shortest distance and pick up the smallest one.
- The formula for SSE:
  $SSE_{C_xC_y} = \frac{1}{2|C_x|}\sum_{x\epsilon C_x}\sum_{y\epsilon C_x} dist(x,y)^2 + \frac{1}{2|C_y|}\sum_{x\epsilon C_y}\sum_{y\epsilon C_y} dist(x,y)^2 + \frac{1}{2|C_x+C_y|}\sum_{x\epsilon C_x}\sum_{y\epsilon C_y} dist(x,y)^2$
- We assume that the SSE is the average of all points in the cluster from the centre for a certain cluster.

## Problem 3

What are **Association Rules**?

- The idea of Association Rules is that it can be defined as a pattern stating the occurrence of one event, by which another event occurs with a certain probability.
- In other words, they are simple If/Else statements which help us discover patterns between unrelated data.
- The interesting part of Association Rules is that they help us learn relationships of objects which are frequently collectively used.
- The goal is to find all sets (only important ones) having *support >minsup* (minimum support).
- Followed by checking which rules cross a certain level of confidence. It means that not all rules generated are important, instead simply pick those rules which have *confidence >minconf.*
- The two most basic and important things to look out while mining Association Rules are **Support** & **Confidence**. Also **Lift** is another alternative, due to the shortcomings of the prior.


Rule: $X \Rightarrow Y$
$Support = \frac{frq(X,Y)}{N}$
$Confidence = \frac{frq(X,Y)}{frq(X)}$
$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$

- Some places where this is used is:
  - Market Basket Analysis.
  - Medical Diagnosis
  - Protein Sequences
  - Census Data

The basic **Apriori Algorithm** is divided into two parts:

- Finding Frequent Itemsets.
  - Generating Candidate Itemsets.
  - Filtering the useful candidates.

Problem 3 continued on next page...

- Mining/Discovering Unknown Patterns from frequent itemsets.

1. How to find the Frequent Itemsets?
    - The very basic need is to specify a **Support level**.
    - Next step consists of actually scanning through the input transaction set and only collecting that cross the threshold of Support with 1 - item set in mind. In other words, eliminate those who do not matter and contribute towards forming interesting patterns.
    - Once we filter based on the support level, we start building candidate sets of k - items.
    - This is gradually done by combining 1 - item sets to form 2 item sets, moving onto three and so on until you do not find an empty set since none crossed the support threshold and k-1 set is returned.
    - So the set generated as C1 (Candidate 1) is filtered to become L1 or F1 (Frequent Sets), which in turn becomes C2 followed by filtering to become C3 and so on.
    - The preprocess.py contains the program to convert datasets to sparse matrix.
    - This is done with the help of two methods:

    $$F_k - 1 * F_k - 1$$

    $$F_k - 1 * F_1$$

    - Implementation is done along with apriori in apriori.py (Answer to A part of question).
    - The First method generates lesser number of subsets compared to the latter one since the number of combinations are sorted lexicographically and then the combinations are made for those item sets who have k-2 elements in common at each level of iteration. So two generate a k=4, it's first two itemsets should have been the same to form a possible combination.
    - The Second method generates more item sets because of the fact even though they are sorted they are combined with Candidate 1 set. There is a better chance of duplication generation here rather than the prior method.
    - Once we find the possible combinations that make are possible, we then move to remove that do not make sense or rather that do not pass the **support threshold**.
    - Once we are through with this, then comes the rule generation. The frequent itemset is passed on to the next step.

2. How to find maximal and closed frequent itemsets?
    - Implementation of C part is done in Misc-Max-Min.py
    - This is simple, starting from the top of the lattice just look for itemsets in the current level to be subsets of the item sets of next level, if they are subset then they are not maximal frequent item set. If they are not subsets they are meant to be maximal frequent itemset.
    - This is simple, starting from the top of the lattice just look for itemsets in the current level to have the same support as that of the item sets of lower level, if they have the same support count then they are not closed frequent item set. If they are not subset they are meant to be closed frequent itemset.
    - All the binarization (B part of the question) for each of the datasets is provided in the code file with the relevant dataset name followed by preprocessing $eg : Nursery_preproccessing$. The Contraceptive dataset has three files, an additional to just convert numerical to categorical.

- The result tells us that the size of Candidate $\geq$ Frequent $\geq$ Closed $\geq$ Maximal. The results below confirm to the principle.
- This is the table for Nuresery dataset `https://archive.ics.uci.edu/ml/datasets/Nursery`

$$F_k - 1 * F_k - 1$$

| Sr. No. | Support Value | Candidate | Frequent | Maximal | Closed |
|---|---|---|---|---|---|
| 1 | 0.1 or 10% | 822 | 166 | 113 | 164 |
| 2 | 0.01 or 1% | 21813 | 6853 | 1683 | 2010 |
| 3 | 0.05 or 5% | 2123 | 600 | 157 | 214 |

- This is the table for Nuresery dataset `https://archive.ics.uci.edu/ml/datasets/Nursery`

$$F_k - 1 * F_1$$

| Sr. No. | Support Value | Candidate | Frequent | Maximal | Closed |
|---|---|---|---|---|---|
| 1 | 0.1 or 10% | 1106 | 166 | 113 | 164 |
| 2 | 0.01 or 1% | 43422 | 6853 | 1683 | 2010 |
| 3 | 0.05 or 5% | 4873 | 600 | 157 | 214 |

- This is the table for Car dataset `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`

$$F_k - 1 * F_1$$

| Sr. No. | Support Value | Candidate | Frequent | Maximal | Closed |
|---|---|---|---|---|---|
| 1 | 0.1 or 10% | 315 | 86 | 43 | 50 |
| 2 | 0.01 or 1% | 5383 | 2291 | 1000 | 1083 |
| 3 | 0.05 or 5% | 1436 | 349 | 127 | 150 |

- This is the table for Car dataset `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`

$$F_k - 1 * F_k - 1$$

| Sr. No. | Support Value | Candidate | Frequent | Maximal | Closed |
|---|---|---|---|---|---|
| 1 | 0.1 or 10% | 475 | 86 | 43 | 50 |
| 2 | 0.01 or 1% | 7104 | 2291 | 1000 | 1083 |
| 3 | 0.05 or 5% | 1838 | 349 | 127 | 150 |

- This is the table for Contraceptive dataset `https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice`

$$F_k - 1 * F_k - 1$$

| Sr. No. | Support Value | Candidate | Frequent | Maximal | Closed |
|---|---|---|---|---|---|
| 1 | 0.1 or 10% | 1937 | 767 | 166 | 406 |
| 2 | 0.01 or 1% | 37634 | 21819 | 539 | 834 |
| 3 | 0.05 or 5% | 6243 | 2726 | 442 | 597 |

- This is the table for Contraceptive dataset `https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice`

$$F_k - 1 * F_1$$

Problem 3 continued on next page. . .

| Sr. No. | Support Value | Candidate | Frequent | Maximal | Closed |
|---|---|---|---|---|---|
| 1 | 0.1 or 10% | 4962 | 767 | 166 | 407 |
| 2 | 0.01 or 1% | 66771 | 21819 | 539 | 834 |
| 3 | 0.05 or 5% | 13691 | 2726 | 442 | 597 |

3. Now we find the Association Rules, how do we do that?

   - It is pretty straight forward. We now know the number of frequent itemsets and the sets. All we need to see is that the frequent itemsets generated, how worthy are they to us in terms of association.
   - We now generate rules. The first step is to enumerate and split each item in the frequent itemset. Once we know each individual element, we go on to merge them again. But why merge when you are splitting? We need to leverage each combination and see that whether if that combination does make sense at all or not?
   - How do we make sense? We can make sense by calculation of confidence of lift. Confidence and Lift are the two different ways to evaluate the rule, i.e. by lift or checking of minimum confidence threshold.
   - While coding we check if there are more than two sets, we look to merge them.
   - There are two parts of a rule *Antecedent*, which is on the left and *consequent*, which is on the right.
   - That being said we calculate for each support, various confidence values.
   - Confidence based pruning refers to removal of rule based on the idea that a tree branch should not be generated in case if that rule does not qualify for the minimum confidence threshold.
   - This means that superset of the current rule branch cannot have more confidence and hence does not qualify for as a good rule.
   - Part D, E and F are implemented over a number of files. But just running the main with the dataset should provide you with the results as per your customisation.
   - D Part:

     – This is the table for Nuresery dataset `https://archive.ics.uci.edu/ml/datasets/Nursery`

| Sr. No. | Support Value | Confidence | Rules Generated | Pruning |
|---|---|---|---|---|
| 1 | 0.01 or 1% | 0.25 | 41235 | 6853 |
| 2 | 0.01 or 1% | 0.5 | 33861 | 693 |
| 3 | 0.01 or 1% | 0.75 | 31012 | 4 |
| 4 | 0.10 or 10% | 0.25 | 341 | 289 |
| 5 | 0.10 or 10% | 0.5 | 337 | 16 |
| 6 | 0.10 or 5% | 0.75 | 279 | 3 |
| 7 | 0.25 or 25% | 0.25 | 18 | 3 |
| 8 | 0.25 or 25% | 0.5 | 7 | 3 |
| 9 | 0.25 or 25% | 0.75 | 7 | 3 |

     – This is the table for Car dataset `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`

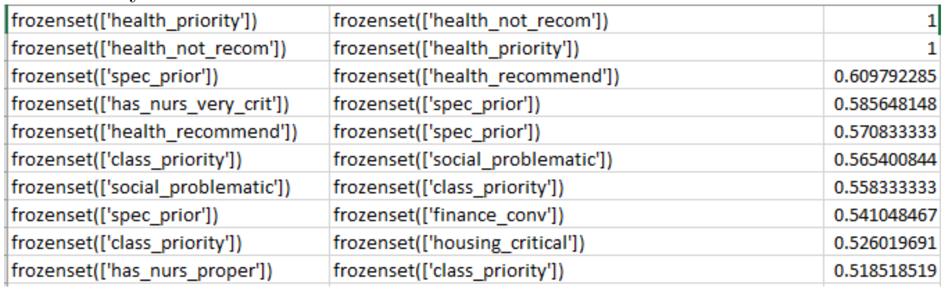
| Sr. No. | Support Value | Confidence | Rules Generated | Pruning |
|---|---|---|---|---|
| 1 | 0.01 or 1% | 0.25 | 9036 | 1593 |
| 2 | 0.01 or 1% | 0.5 | 9010 | 43 |
| 3 | 0.01 or 1% | 0.75 | 9010 | 8 |
| 4 | 0.10 or 10% | 0.25 | 137 | 118 |
| 5 | 0.10 or 10% | 0.5 | 137 | 22 |
| 6 | 0.10 or 5% | 0.75 | 137 | 7 |
| 7 | 0.25 or 25% | 0.25 | 6 | 6 |
| 8 | 0.25 or 25% | 0.5 | 6 | 3 |
| 9 | 0.25 or 25% | 0.75 | 6 | 3 |

– This is the table for Contraceptive dataset `https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice`

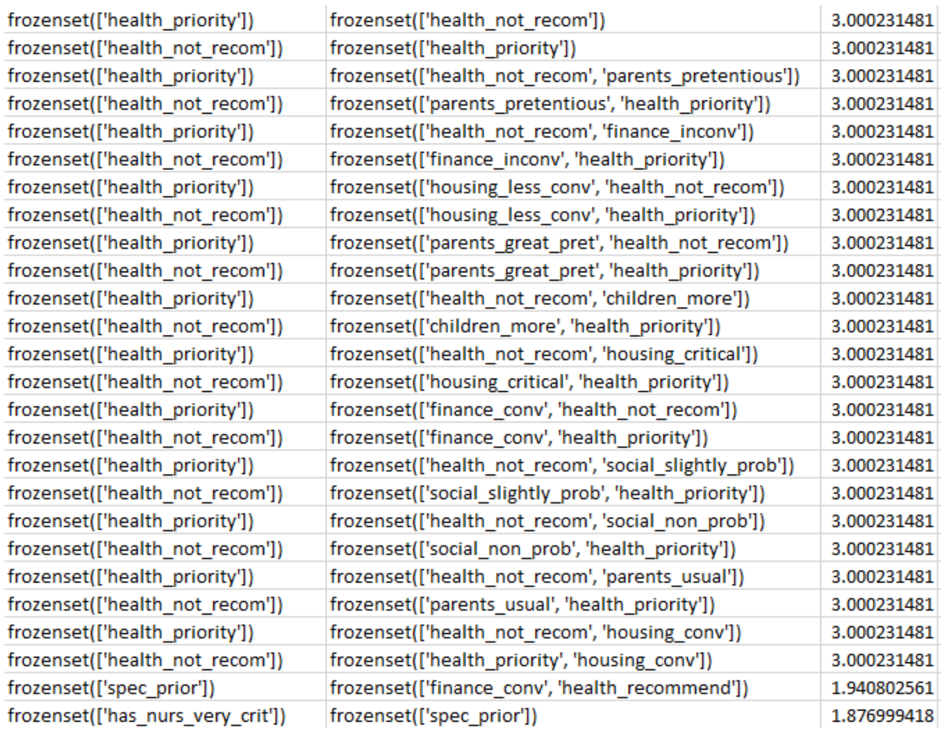
| Sr. No. | Support Value | Confidence | Rules Generated | Pruning |
|---|---|---|---|---|
| 1 | 0.01 or 1% | 0.25 | 4551 | 3679 |
| 2 | 0.01 or 1% | 0.5 | 3474 | 1270 |
| 3 | 0.01 or 1% | 0.75 | 3190 | 311 |
| 4 | 0.10 or 10% | 0.25 |  | 289 |
| 5 | 0.10 or 10% | 0.5 | 255 | 143 |
| 6 | 0.10 or 5% | 0.75 | 243 | 39 |
| 7 | 0.25 or 25% | 0.25 | 279 | 279 |
| 8 | 0.25 or 25% | 0.5 | 255 | 143 |
| 9 | 0.25 or 25% | 0.75 | 243 | 63 |

- Here are the top 10s and the lift execution screenshots in sorted order.

  – For Nursery Dataset

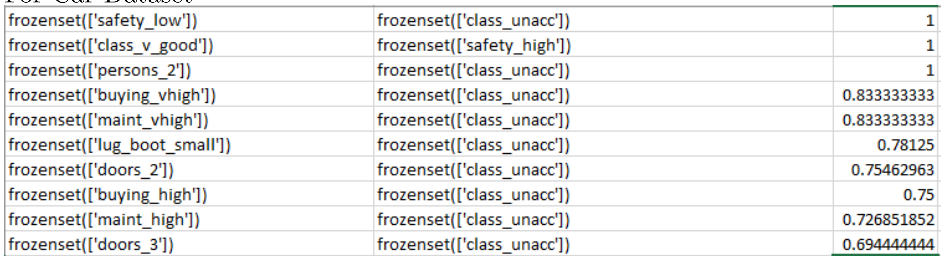


| | | |
|---|---|---|
| frozenset(['health_priority']) | frozenset(['health_not_recom']) | 1 |
| frozenset(['health_not_recom']) | frozenset(['health_priority']) | 1 |
| frozenset(['spec_prior']) | frozenset(['health_recommend']) | 0.609792285 |
| frozenset(['has_nurs_very_crit']) | frozenset(['spec_prior']) | 0.585648148 |
| frozenset(['health_recommend']) | frozenset(['spec_prior']) | 0.570833333 |
| frozenset(['class_priority']) | frozenset(['social_problematic']) | 0.565400844 |
| frozenset(['social_problematic']) | frozenset(['class_priority']) | 0.558333333 |
| frozenset(['spec_prior']) | frozenset(['finance_conv']) | 0.541048467 |
| frozenset(['class_priority']) | frozenset(['housing_critical']) | 0.526019691 |
| frozenset(['has_nurs_proper']) | frozenset(['class_priority']) | 0.518518519 |

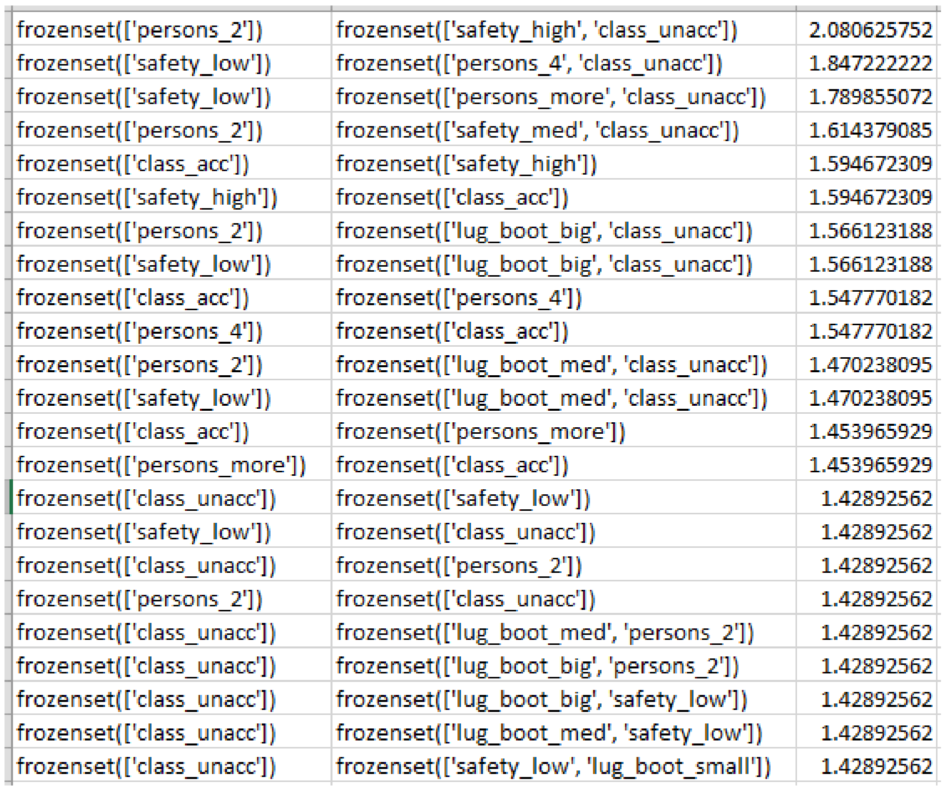


| | | |
|---|---|---|
| frozenset(['health_priority']) | frozenset(['health_not_recom']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['health_not_recom', 'parents_pretentious']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['parents_pretentious', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['health_not_recom', 'finance_inconv']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['finance_inconv', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['housing_less_conv', 'health_not_recom']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['housing_less_conv', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['parents_great_pret', 'health_not_recom']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['parents_great_pret', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['health_not_recom', 'children_more']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['children_more', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['health_not_recom', 'housing_critical']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['housing_critical', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['finance_conv', 'health_not_recom']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['finance_conv', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['health_not_recom', 'social_slightly_prob']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['social_slightly_prob', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['health_not_recom', 'social_non_prob']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['social_non_prob', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['health_not_recom', 'parents_usual']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['parents_usual', 'health_priority']) | 3.000231481 |
| frozenset(['health_priority']) | frozenset(['health_not_recom', 'housing_conv']) | 3.000231481 |
| frozenset(['health_not_recom']) | frozenset(['health_priority', 'housing_conv']) | 3.000231481 |
| frozenset(['spec_prior']) | frozenset(['finance_conv', 'health_recommend']) | 1.940802561 |
| frozenset(['has_nurs_very_crit']) | frozenset(['spec_prior']) | 1.876999418 |

– For Car Dataset

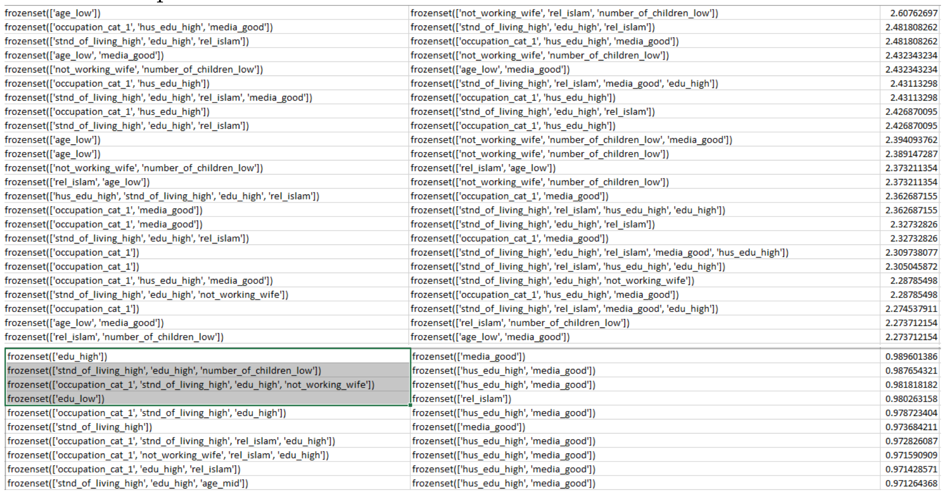


| | | |
|---|---|---|
| frozenset(['safety_low']) | frozenset(['class_unacc']) | 1 |
| frozenset(['class_v_good']) | frozenset(['safety_high']) | 1 |
| frozenset(['persons_2']) | frozenset(['class_unacc']) | 1 |
| frozenset(['buying_vhigh']) | frozenset(['class_unacc']) | 0.833333333 |
| frozenset(['maint_vhigh']) | frozenset(['class_unacc']) | 0.833333333 |
| frozenset(['lug_boot_small']) | frozenset(['class_unacc']) | 0.78125 |
| frozenset(['doors_2']) | frozenset(['class_unacc']) | 0.75462963 |
| frozenset(['buying_high']) | frozenset(['class_unacc']) | 0.75 |
| frozenset(['maint_high']) | frozenset(['class_unacc']) | 0.726851852 |
| frozenset(['doors_3']) | frozenset(['class_unacc']) | 0.694444444 |



| | | |
|---|---|---|
| frozenset(['persons_2']) | frozenset(['safety_high', 'class_unacc']) | 2.080625752 |
| frozenset(['safety_low']) | frozenset(['persons_4', 'class_unacc']) | 1.847222222 |
| frozenset(['safety_low']) | frozenset(['persons_more', 'class_unacc']) | 1.789855072 |
| frozenset(['persons_2']) | frozenset(['safety_med', 'class_unacc']) | 1.614379085 |
| frozenset(['class_acc']) | frozenset(['safety_high']) | 1.594672309 |
| frozenset(['safety_high']) | frozenset(['class_acc']) | 1.594672309 |
| frozenset(['persons_2']) | frozenset(['lug_boot_big', 'class_unacc']) | 1.566123188 |
| frozenset(['safety_low']) | frozenset(['lug_boot_big', 'class_unacc']) | 1.566123188 |
| frozenset(['class_acc']) | frozenset(['persons_4']) | 1.547770182 |
| frozenset(['persons_4']) | frozenset(['class_acc']) | 1.547770182 |
| frozenset(['persons_2']) | frozenset(['lug_boot_med', 'class_unacc']) | 1.470238095 |
| frozenset(['safety_low']) | frozenset(['lug_boot_med', 'class_unacc']) | 1.470238095 |
| frozenset(['class_acc']) | frozenset(['persons_more']) | 1.453965929 |
| frozenset(['persons_more']) | frozenset(['class_acc']) | 1.453965929 |
| frozenset(['class_unacc']) | frozenset(['safety_low']) | 1.42892562 |
| frozenset(['safety_low']) | frozenset(['class_unacc']) | 1.42892562 |
| frozenset(['class_unacc']) | frozenset(['persons_2']) | 1.42892562 |
| frozenset(['persons_2']) | frozenset(['class_unacc']) | 1.42892562 |
| frozenset(['class_unacc']) | frozenset(['lug_boot_med', 'persons_2']) | 1.42892562 |
| frozenset(['class_unacc']) | frozenset(['lug_boot_big', 'persons_2']) | 1.42892562 |
| frozenset(['class_unacc']) | frozenset(['lug_boot_big', 'safety_low']) | 1.42892562 |
| frozenset(['class_unacc']) | frozenset(['lug_boot_med', 'safety_low']) | 1.42892562 |
| frozenset(['class_unacc']) | frozenset(['safety_low', 'lug_boot_small']) | 1.42892562 |

- For Contraception Dataset



| | | |
|---|---|---|
| frozenset(['age_low']) | frozenset(['not_working_wife', 'rel_islam', 'number_of_children_low']) | 2.60762697 |
| frozenset(['occupation_cat_1', 'hus_edu_high', 'media_good']) | frozenset(['stnd_of_living_high', 'edu_high', 'rel_islam']) | 2.481808262 |
| frozenset(['stnd_of_living_high', 'edu_high', 'rel_islam']) | frozenset(['occupation_cat_1', 'hus_edu_high', 'media_good']) | 2.481808262 |
| frozenset(['age_low', 'media_good']) | frozenset(['not_working_wife', 'number_of_children_low']) | 2.432343234 |
| frozenset(['not_working_wife', 'number_of_children_low']) | frozenset(['age_low', 'media_good']) | 2.432343234 |
| frozenset(['occupation_cat_1', 'hus_edu_high']) | frozenset(['stnd_of_living_high', 'rel_islam', 'media_good', 'edu_high']) | 2.43113298 |
| frozenset(['stnd_of_living_high', 'edu_high', 'rel_islam', 'media_good']) | frozenset(['occupation_cat_1', 'hus_edu_high']) | 2.43113298 |
| frozenset(['occupation_cat_1', 'hus_edu_high']) | frozenset(['stnd_of_living_high', 'edu_high', 'rel_islam']) | 2.426870095 |
| frozenset(['stnd_of_living_high', 'edu_high', 'rel_islam']) | frozenset(['occupation_cat_1', 'hus_edu_high']) | 2.426870095 |
| frozenset(['age_low']) | frozenset(['not_working_wife', 'number_of_children_low', 'media_good']) | 2.394093762 |
| frozenset(['age_low']) | frozenset(['not_working_wife', 'number_of_children_low']) | 2.389147287 |
| frozenset(['not_working_wife', 'number_of_children_low']) | frozenset(['rel_islam', 'age_low']) | 2.373211354 |
| frozenset(['rel_islam', 'age_low']) | frozenset(['not_working_wife', 'number_of_children_low']) | 2.373211354 |
| frozenset(['hus_edu_high', 'stnd_of_living_high', 'edu_high', 'rel_islam']) | frozenset(['occupation_cat_1', 'media_good']) | 2.362687155 |
| frozenset(['occupation_cat_1', 'media_good']) | frozenset(['stnd_of_living_high', 'rel_islam', 'hus_edu_high', 'edu_high']) | 2.362687155 |
| frozenset(['occupation_cat_1', 'media_good']) | frozenset(['stnd_of_living_high', 'edu_high', 'rel_islam']) | 2.32732826 |
| frozenset(['stnd_of_living_high', 'edu_high', 'rel_islam']) | frozenset(['occupation_cat_1', 'media_good']) | 2.32732826 |
| frozenset(['occupation_cat_1']) | frozenset(['stnd_of_living_high', 'edu_high', 'rel_islam', 'media_good', 'hus_edu_high']) | 2.309738077 |
| frozenset(['occupation_cat_1']) | frozenset(['stnd_of_living_high', 'rel_islam', 'hus_edu_high', 'edu_high']) | 2.305045872 |
| frozenset(['occupation_cat_1', 'hus_edu_high', 'media_good']) | frozenset(['stnd_of_living_high', 'edu_high', 'not_working_wife']) | 2.28785498 |
| frozenset(['stnd_of_living_high', 'edu_high', 'not_working_wife']) | frozenset(['occupation_cat_1', 'hus_edu_high', 'media_good']) | 2.28785498 |
| frozenset(['occupation_cat_1']) | frozenset(['stnd_of_living_high', 'rel_islam', 'media_good', 'edu_high']) | 2.274537911 |
| frozenset(['age_low', 'media_good']) | frozenset(['rel_islam', 'number_of_children_low']) | 2.273712154 |
| frozenset(['rel_islam', 'number_of_children_low']) | frozenset(['age_low', 'media_good']) | 2.273712154 |

| | | |
|---|---|---|
| frozenset(['edu_high']) | frozenset(['media_good']) | 0.989601386 |
| frozenset(['stnd_of_living_high', 'edu_high', 'number_of_children_low']) | frozenset(['hus_edu_high', 'media_good']) | 0.987654321 |
| frozenset(['occupation_cat_1', 'stnd_of_living_high', 'edu_high', 'not_working_wife']) | frozenset(['hus_edu_high', 'media_good']) | 0.981818182 |
| frozenset(['edu_low']) | frozenset(['rel_islam']) | 0.980263158 |
| frozenset(['occupation_cat_1', 'stnd_of_living_high', 'edu_high']) | frozenset(['hus_edu_high', 'media_good']) | 0.978723404 |
| frozenset(['stnd_of_living_high']) | frozenset(['media_good']) | 0.973684211 |
| frozenset(['occupation_cat_1', 'stnd_of_living_high', 'rel_islam', 'edu_high']) | frozenset(['hus_edu_high', 'media_good']) | 0.972826087 |
| frozenset(['occupation_cat_1', 'not_working_wife', 'rel_islam', 'edu_high']) | frozenset(['hus_edu_high', 'media_good']) | 0.971590909 |
| frozenset(['occupation_cat_1', 'edu_high', 'rel_islam']) | frozenset(['hus_edu_high', 'media_good']) | 0.971428571 |
| frozenset(['stnd_of_living_high', 'edu_high', 'age_mid']) | frozenset(['hus_edu_high', 'media_good']) | 0.971264368 |

## Problem 4

**Paper A**
**The idea of the paper**

The author mentions the study of clustering in this paper. He proposes the idea of a unified clustering of the objects and the difficulty in formulating a proposition which can be applied to all objects. He disproves the coexistence of three propositions because he then claims that there exist no such function which can satisfy those three properties of Scale Invariance, Richness and Consistency altogether. He displays the flaws in some algorithms which limit us to achieve the three aforementioned properties together.

- **Introduction**
  The congregation of similar objects can be defined as Clusters. In other words, points having similar properties tend to be closer to each other than points they are less similar to forming clusters. Similar points in one cluster are less similar to points in other clusters. But that being said it should not be the only idea or aspect affecting the formation of a cluster, various accounts need to be considered while forming a cluster. The various axiomatic principles have been taken into account and their relation to the cluster properties that have been mentioned.
- **The Impossibility Theorem**
  The author talks about the restriction of distance metric which maybe distance and the triangular inequality property of a metric. Though he does not actually use it, it is of significant importance in understanding the other three properties. He mainly is talking about three properties:

  1. Scale Invariance - The author checks the sensitivity of a function. He claims that the quantity of measure taken should not affect the result.

  $$f(d) = f(\alpha d)$$

  2. Richness - The author here wants to say that the reconstruction of the output should be generated or can be generated by any partition and should be equally probable to be selected.
  3. Consistency - This is one of the important and impactful property. The property talks about the consistency. It says that when intra cluster distance is appreciated and inter cluster distance is depreciated we should still obtain the same result. The author defines new variables $d$ & $d'$, where $d$ is the original distance and then $d'$ is the distance which is calculated after manipulation.
- **Claims**
  The author now claims that if the n greater than or equal 2 , then there exists no function that could satisfy the aforementioned properties. Before he proves this, the author shows the interrelation of these properties and shows that any two properties can be satisfied at one time but not three. The author takes into consideration the single linkage properties which says clusters can be represented as weighted graphs and sub graphs and defines three terminating conditions:

  1. K - Cluster Property - When we achieve a graph of K connected components we terminate.
  2. Distance R Property - Weights more than R should be avoided.
  3. Scaling - alpha Property - P Indicates the max pairwise distance, weights of about scaled P is appended.
- Hence the conclusion based on the above mentioned terminating condition we confirm that only any two rules can be satisfied.

Problem 4 continued on next page. . .

- The various combinations that are possible:
  - When K is greater than or equal to 1 and for any n greater than equal to K will only satisfy Properties 1 & 3.
  - Any positive $\alpha$ along with scale $\alpha$ stopping condition satisfy 1 & 2.
  - Finally for R greater than 0, and n greater than or equal to 2, 2 & 3 are satisfied.
- **Anitchains**
  In order to prove it in a much consistent and solid manner we use a set of special refinement cluster. Let us understand by an example. Let us say that the resultant cluster is $A$ and $B$, then there exists a partition cluster $A'$ which will be a $A' \subseteq A$ and also $B' \subseteq A$. Once we prove this the author simply goes ahead and proves the proposition. He finds the flaws in the equation mentioned above. That is say if Scale and Consistency property are satisfied then the range function will form an **Antichain**. Similarly other two propositions are further strengthened.
- **Centroid Based Clustering and Consistency**
  The author in this part of the paper claims to have said that the **centroid** based algorithms do not or cannot satisfy the Consistency Property. He proposes to us the idea of a new method known as $(k, g)$-based centroid clustering method. This method does the selection of centroids from a set of already existing points which is then followed by assigning points to the closest centroid forming clusters.
- **Relaxing Properties**
  Here the author actually wants us to understand the idea of relaxation so that all the properties can be satisfied. He introduces the idea of Refinement Consistency saying that if $d'$ being a transformation of $f(d)$, then $f(d')$ should be a refined version of $f(d)$.
- **My Understanding**
  As far as my knowledge extends I have come to the conclusion that the Consistency property has a major role to play. If we could somehow introduce a function and relax the consistency property a little bit, we sure would be able to satisfy the other properties. The size of a cluster and the number of clusters also play a major role in the idea of satisfying all the properties (which is mentioned in the next paper), keeping in mind all this if only the author could expand his horizon and extend to something more than Clustering functions to find a solution to satisfy the properties.

**Paper B**

**The idea of the paper**

This paper is in relation to the first paper mentioning the idea of **Impossibility Theorem**. The author discusses Cluster Qualities and opposes the claims made by in the first paper. She claims that the instead of defining the properties, let us work with the cluster quality without leading to any sort of inconsistencies. Clustering Quality can be best understood as a function which returns a non negative number representing how good the algorithm performs. Also she claims that impossibility is not an inherent feature but due to various assumed propositions it has become so.

- **Introduction**
  In this paper the author introduces Clustering Quality Measure (CQM) which is independent of any specific algorithm and which she claims is much more general. This introduces the idea of comparing the result obtained by one algorithm with the result obtained by another, which gives us a common

Problem 4 continued on next page. . .

platform to compare them. Running time, as claimed by the author is Polynomial Time. The author then discusses the axioms and their weakness. The discussion takes into account as to on what grounds were the axioms proposed and how the consistency axiom is not good in terms of clustering functions. The author then on basis of the three basic properties prepares the first CQM after reformulating them.

- **Relative Margin**
  The author begins with defining the idea of Relative Point Margin. She says that relative margin is a mean of distances, where in when a point is selected we also select the closest center points followed by taking a ratio between them which is then appended to form a complete representative set. The realisation of a cluster to be good is to have a lower value of the ratio. The author then talks about the proposition by stating 3 lemmas in relation to properties.

- **Soundness and Completeness**
  The author in this part defines Completeness and Soundness which are meant to be compatible with the axioms. Soundness is something that says there should be consistency all the axioms, in other words it says that all the elements should be compatible with all the axioms. The term Completeness says that all the properties used by objects should be inherently followed by the axioms. She goes on to say that CQM tends to comply to the soundness and completeness properties, the clustering functions may fail.

- **Isomorphism Invariance**
  This axiom is interesting as it mentions the treating of each cluster with the same respect. Say two clusters are isomorphic if and only if there is preservation of distance between them. She says that there is indifference because a consistent set of CQM satisfies Isomorphism Invariance, Scale Invariance, Consistency and Richness.

- **More about CQM**

  – Weakest Link - The idea here is to see that points that do not belong to the same cluster are loosely packed while those in the same are tightly coupled with links among them. It is moreover the extremes or the longest links where in the emphasis is given to the longest links. Then the maximum value is selected and then is divided by the smallest distance between two clusters.

  – Additive Margin - Another example would be this, though it is a little different as it calculate the difference instead of the ratio as it considers the distance between two of its most closest cluster centres. It is the average of of Additive Point Margin averaging the intra cluster distance.

- The author in the last part talks about the computational complexity. The author claims that the function is in terms of K and has polynomial complexity as $O(n^{k+1})$, but if centres are pre specified then they take lesser times, such that Relative Margin takes $O(nk)$ while Additive Margin takes $O(n^2)$ and Weakest Link takes $O(n^3)$. As mentioned earlier that number of clusters can play a major role while it is not discussed here. The authors the objective functions will fail to comply with the Scale Invariance and Richness but introduction of a new normalisation technique as said by the author can be used to overcome the issues. Another interesting thing the author mentions refinement preferring and coarsening preferring (they fail the richness property) which are quite similar to the previous papers annitchains. As they fail the richness property we again need to define the number of clusters while testing the quality of clusters using refinement.

- **My Understanding** The author convincingly disproves the impossibility theorem and has with a very cogent study of how she went about doing that. The solution provided by the author in terms of CQM is a valid point but each method has their disadvantages. The idea of dependence on the number of

Problem 4 continued on next page. . .

clusters is the focal point and not always will CQM come through, I still would not be convinced that this method will work in all environments and on all objects and hence would still search for a method which would work ubiquitously.

References

- https://www.researchgate.net/publication/238525379_Association_rule_mining-_Applications_in_various_areas
- https://www.youtube.com/watch?v=RHkvnRemaLE&index=3&list=PLVOXKA8fjRuvTVJt_n1rkRl6n3sGcyY6a
- http://aimotion.blogspot.com/2013/01/machine-learning-and-data-mining.html
- http://www2.ift.ulaval.ca/~chaib/IFT-4102-7025/public_html/Fichiers/Machine_Learning_in_Action.pdf
- http://papers.nips.cc/paper/3491-measures-of-clustering-quality-a-working-set-of-axion pdf
- http://www.cs.cornell.edu/home/kleinber/nips15.pdf
- https://ccrma.stanford.edu/workshops/mir2009/references/ROCintro.pdf