REPORT

Hyper clique Pattern Discovery

Following is the list of work performed for the Hyper clique pattern Discovery paper :

- Implemented Hyper clique Miner Algorithm (code attached)
- Proof of correctness using association pattern mining examples
- Executing algorithm on **PUMSB dataset**
- Examining the number of patterns generated by hyper clique miner for various minimum support threshold
- Examining the execution time of hyper clique miner for various minimum support threshold
- Examining the number of patterns for various number of attributes
- Examining the execution time for various number of attributes (hyper clique miners)
- Conclusions drawn

Hyper clique Miner Algorithm Implementation

Below is the list of functions used in the algorithm (Complete code has been attached):

def calc_hc(item) : This function calculates h-confidence for a given item set.

def calc_sup(item) : This function calculates the value of support for a given item.

def aprioriGen(Lk, k) : This function returns Ck+1 from the given Lk.

def antimonotone() : This function returns patterns after pruning using anti-monotone property.

def cross_support() : This function returns patterns after pruning using cross-support property.

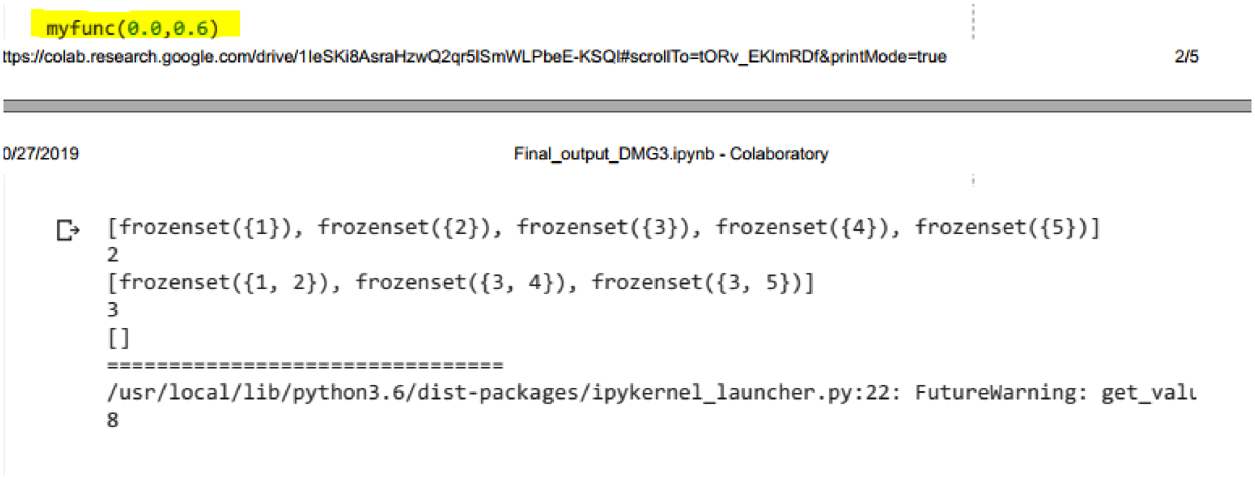
Correctness Proof for the algorithm implementation using association pattern mining example

I have taken the example given on page number 19 of the research paper.

TID	Items
1	1
2	2
3	3, 4
4	1, 2
5	1, 2
6	1, 2
7	1, 2, 3, 4, 5
8	1
9	2
10	3, 5

Minimum support =0.0

h-confidence threshold = 0.6



```
myfunc(0.0,0.6)
```

ttps://colab.research.google.com/drive/1IeSKi8AsraHzwQ2qr5lSmWLPbeE-KSQl#scrollTo=tORv_EKlmRDf&printMode=true 2/5

0/27/2019 Final_output_DMG3.ipynb - Colaboratory

```
[frozenset({1}), frozenset({2}), frozenset({3}), frozenset({4}), frozenset({5})]
2
[frozenset({1, 2}), frozenset({3, 4}), frozenset({3, 5})]
3
[]
================================
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:22: FutureWarning: get_valu
8
```

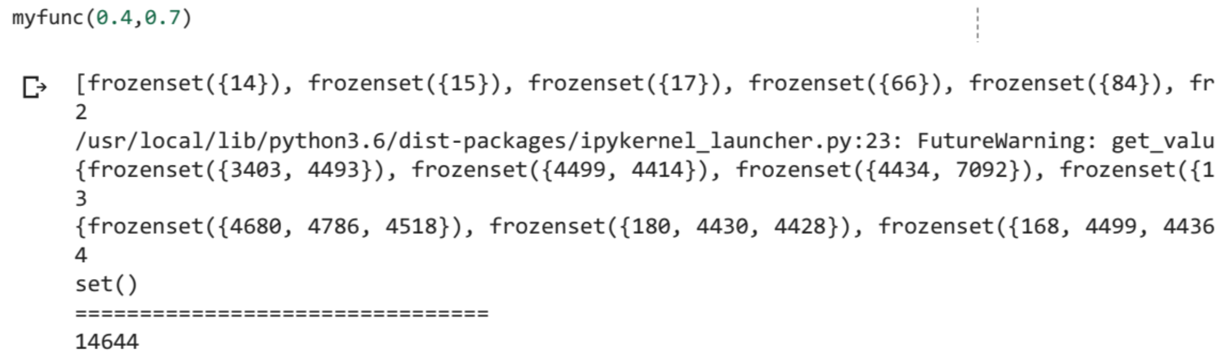
Please refer **Example_execution.pdf** attached for the detailed implementation.

Executing Algorithm code on PUMSB dataset

Since PUMSB dataset was very big , I have done sampling (2k rows) of the given data and executed code for the same .

- Min_sup=0.4
- h-confidence_threshold=0.7
- Total number of patterns found after pruning =14644

Output Screen shot :

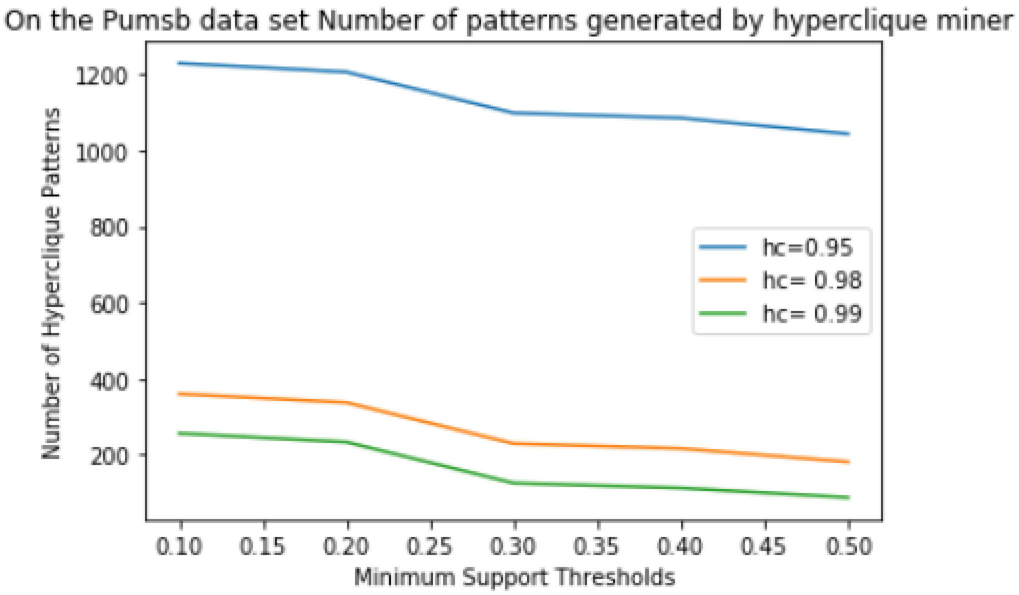


```
myfunc(0.4,0.7)

[frozenset({14}), frozenset({15}), frozenset({17}), frozenset({66}), frozenset({84}), fr
2
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:23: FutureWarning: get_valu
{frozenset({3403, 4493}), frozenset({4499, 4414}), frozenset({4434, 7092}), frozenset({1
3
{frozenset({4680, 4786, 4518}), frozenset({180, 4430, 4428}), frozenset({168, 4499, 4436
4
set()
================================
14644
```

NOTE : Please refer **PUMSB_output.txt** file attached for the generated patterns and **Final_output_PUSMB.pdf** for the complete code execution.

Examining the number of patterns generated by hyper clique miner for various minimum support threshold

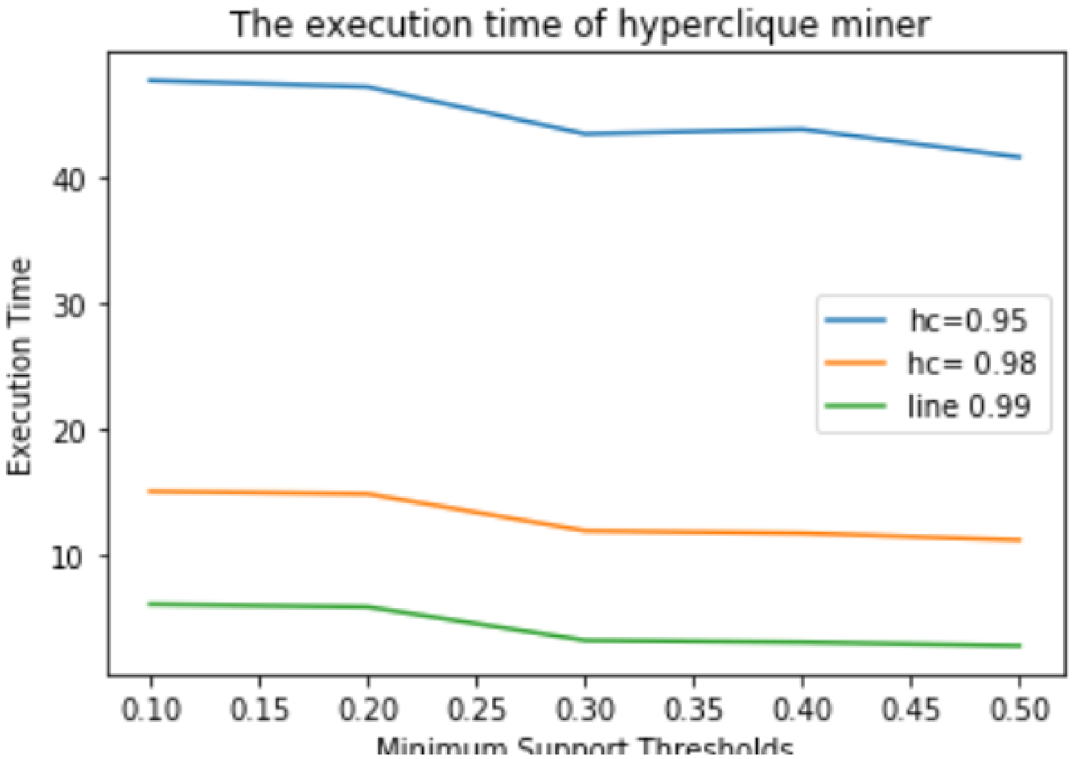


Inferences Drawn :

- Increasing the value of h confidence threshold results in lesser number of hyper clique patterns.
- *Number of hyper clique patterns decreases with increase of min support values for a* constant h confidence threshold.
- Number of hyper clique patterns significantly increases by slight decrement of h conf threshold value.

Please find the implementation details in **minsup_vs_patterns.pdf** attached.

Examining the execution time of hyper clique miner for various minimum support threshold

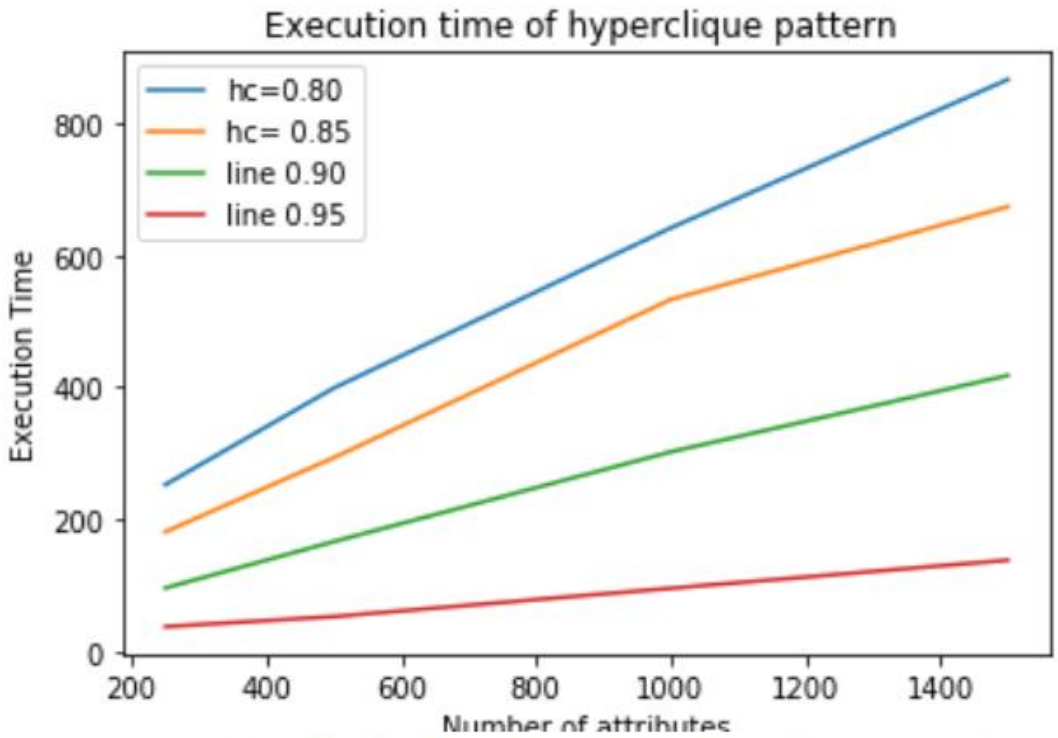


Inferences Drawn:

- Smaller value of H confidence threshold results in larger number of hyper clique patterns. Therefore execution time increases.
- Execution time decreases with increase of min support value for a constant value of h confidence.

Please refer **min_sup_vs executiontime.pdf** for necessary details.

Examining the execution time for various number of attributes (hyper clique miners)

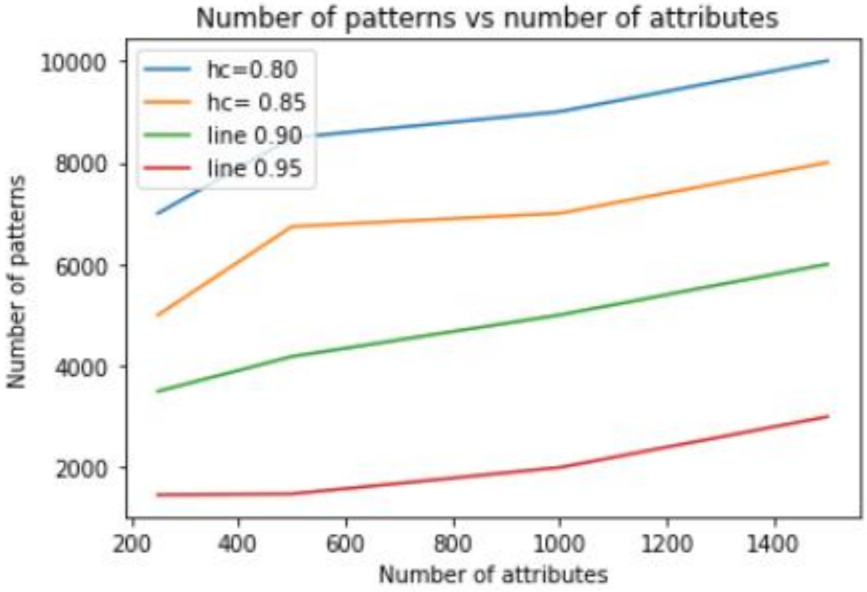


Inferences Drawn

- Execution time increases with increase of number of attributes for a constant h confidence value
- Execution time increases with decrease in h confidence value.

Please refer **number_of_attr_vs_execution_time.pdf** for more details.

Examining the number of patterns for various number of attributes



Inferences Drawn:

- Number of hyper clique patterns increases with increase of number of attributes for a constant h confidence threshold.
- High value of h confidence threshold results in lesser number of hyper clique patterns.

Please refer **attr_vs_number_patterns.pdf** for more details.

Conclusions

- Cross support property (Corollary 1) tested successfully and can be used to avoid generating spurious pattern involving items from different support levels.
- Combination of anti-monotone and cross support properties worked correctly for efficient discovery of hyper clique patterns at low levels of support.
- Using Apriori algorithm may take significant amount of time even for smaller data set and usually gets trapped in low memory issues, hyper clique miner algorithm successfully overcome this problem.
- Though hyper clique miner is efficient yet large datasets requires RAM of larger size hence had to run the code on the sample of the data set.