Hyperclique Pattern Discovery Implementation

Research Paper by –

Hui Xiong (hui@rbs.rutgers.edu)

Pang-Ning Tan (ptan@cse.msu.edu)

Vipin Kumar (kumar@cs.umn.edu)

Problem Description:

Many real time dataset consist of skewed support distribution of items and classic association mining algorithms such as Apriori are not effective in such cases as having too low support threshold will lead to many spurious and undeserved association patterns and having too high support threshold will lead to miss of various important association patterns.

So there is another approach used in terms of H-confidence to generate association patterns in above explained scenario.

Some terms and formulae Used:

DEFINITION 1. *The* ***h-confidence*** *of an itemset* $P = \{i_1, i_2, \cdots, i_m\}$ *is defined as follows:*

$$hconf(P) = \min\{conf\{i_1 \rightarrow i_2, \ldots, i_m\}, conf\{i_2 \rightarrow i_1, i_3, \ldots, i_m\}, \ldots, conf\{i_m \rightarrow i_1, \ldots, i_{m-1}\}\},$$

*where conf follows from the conventional definition of association rule confidence (*Agrawal et al., 1993*).*

DEFINITION 2. *Given a set of items* $I = \{i_1, i_2, \ldots, i_n\}$ *and a minimum h-confidence threshold* h_c*, an itemset* $P \subseteq I$ *is a* **hyperclique pattern** *if and only if* $hconf(P) \geq h_c$.

DEFINITION 4. [*Cross-support Patterns*]: *Given a threshold* t*, a pattern* P *is a cross-support pattern with respect to* t *if* P *contains two items* x *and* y *such that* $\frac{supp(\{x\})}{supp(\{y\})} < t$*, where* $0 < t < 1$.

LEMMA 2. [*Cross-support Property of the h-confidence measure*]: *Any cross-support pattern* P *with respect to a threshold* t *is guaranteed to have* $hconf(P) < t$.

Simplified HyperClique Algorithm:

Hyperclique Miner

Input:

1) a set F of K Boolean feature types F=$\{f_1, f_2, \ldots, f_K\}$

2) a set T of N transactions T=$\{t_1 \ldots t_N\}$, each $t_i \in T$ is a record with K attributes $\{i_1, i_2, \ldots, i_K\}$ taking values in {0, 1}, where the i_p $(1 \leq p \leq K)$ is the Boolean value for the feature type f_p.

3) A user specified minimum h-confidence threshold (h_c)

4) A user specified minimum support threshold (min_supp)

Output:

hyperclique patterns with h-confidence $> h_c$ and support $> min\_supp$

Method:

1) Get size-1 prevalent items

2) **for** the size of itemsets in (2, 3, ... , $K-1$) **do**

3) Generate candidate hyperclique patterns using the *generalized apriori_gen* algorithm

4) Generate hyperclique patterns

5) **end**;

Experimental Details and Observations:

Dataset Used : kosarak

Language Used : python

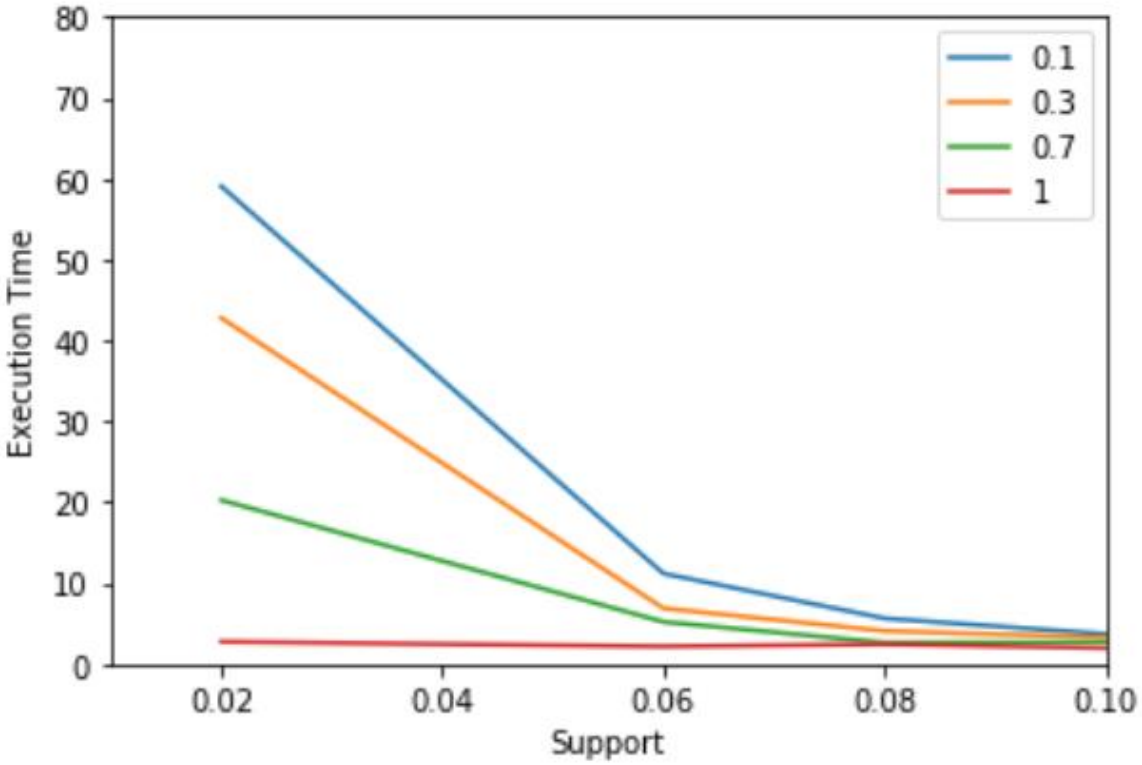
Support and confidence pairs used from following two list with each possible combination acting as

MinSupplist=[0.02,0.06,0.08,0.1] #Support threshold

MinHconflist=[0.1,0.3,0.7,1] #Hconfidence threshold

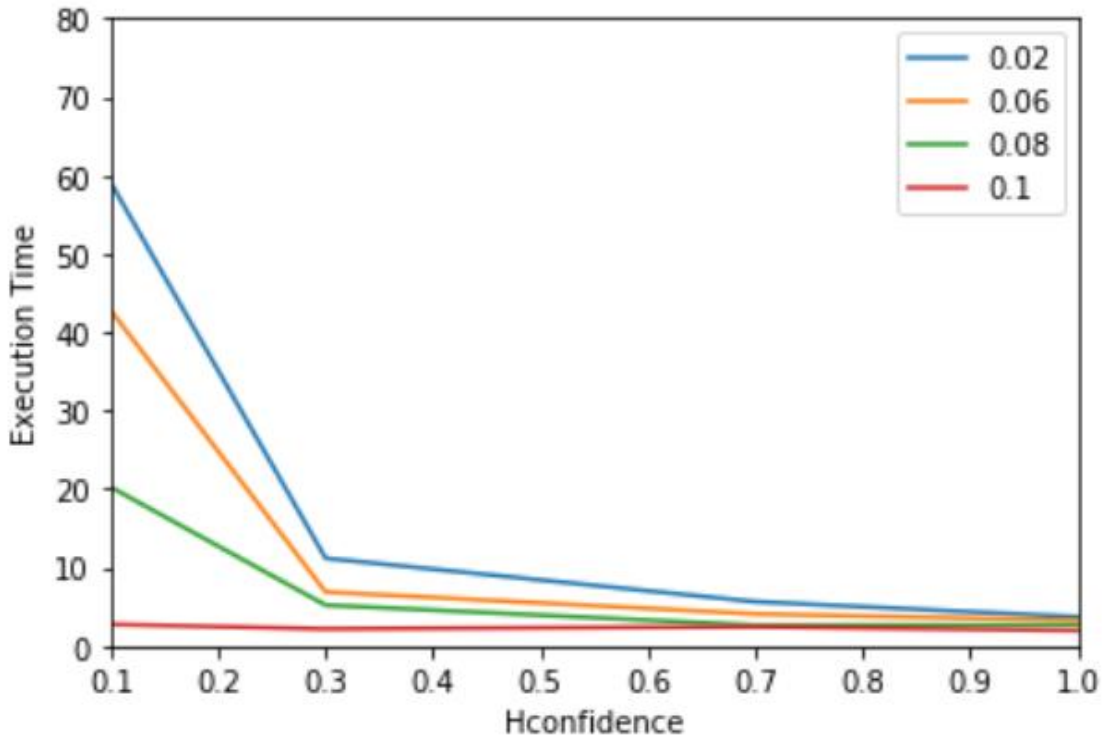
- Graph 1 : Support threshold vs Execution Time

  Execution time in the first set of values of support and hconf threshold is higher but it it decreases very rapidly after the support reached a level of 0.06



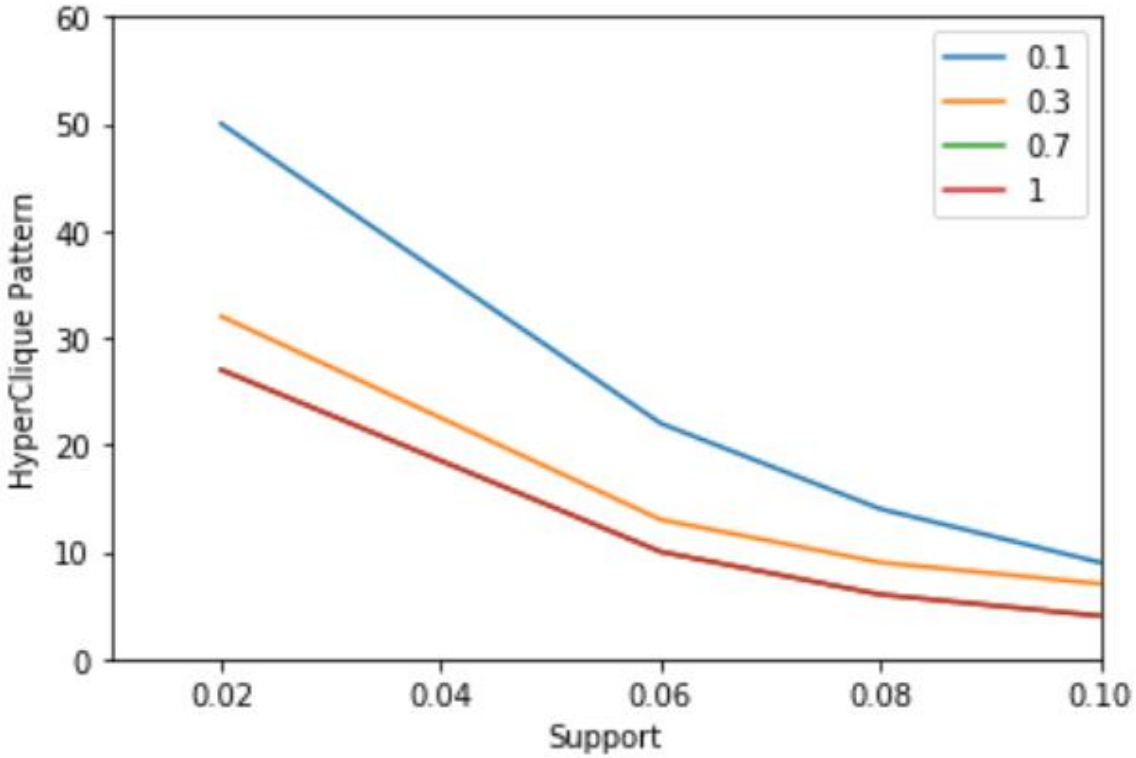
- Graph 2 : Hconfidence threshold vs Execution Time

Execution time in the first set of values of support and hconf threshold is higher but there is sharp decline after hconf reaches 0.3.



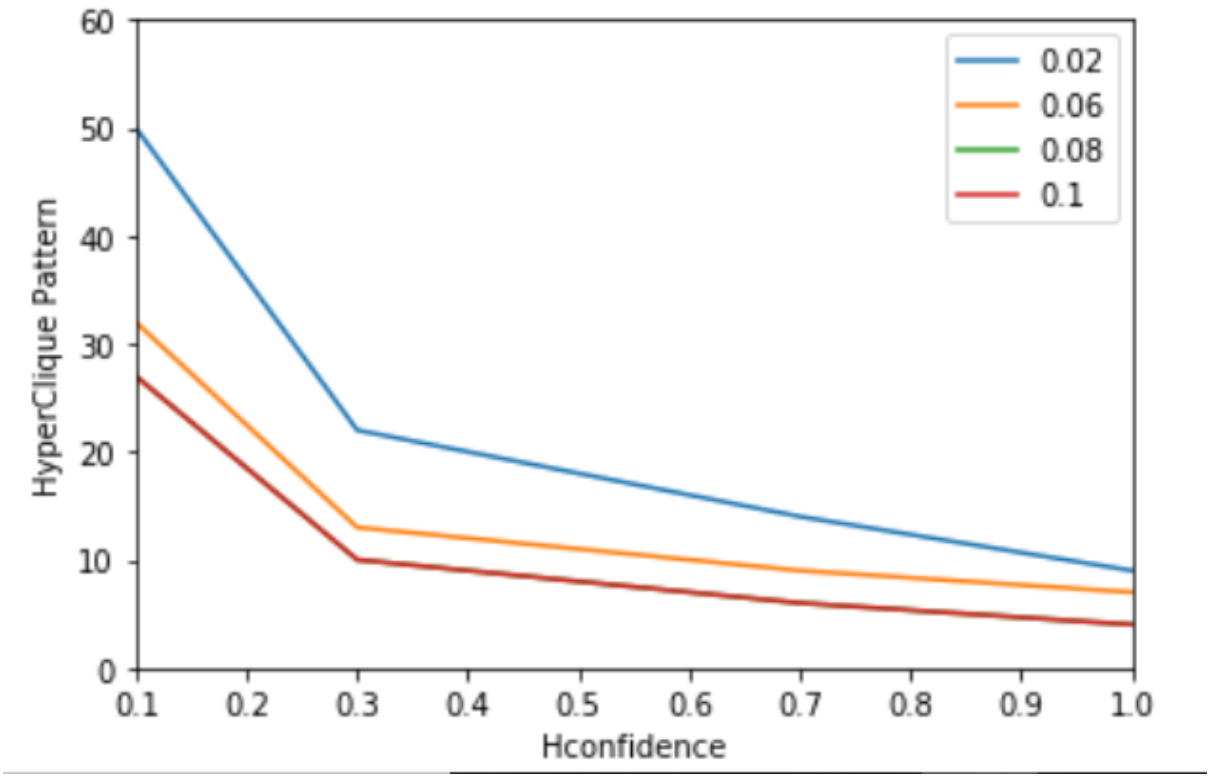
- Graph 3 : Support threshold vs #of Hyper clique patterns

  Clearly, increase in support means less no of Hyperclique patterns



- Graph 4 : Hconfidence threshold vs Hyperclique patterns

  Although, there is decrease in no of Hyperclique patterns but one observation is that decrease is rapid when hconf reaches 0.3



- Graph 5 : Total Hyperclique patterns vs execution time for set of data points

  Execution time has increased from 74s to 180s when dataset size has increased from 25000 to 85000 and there is increase of 3 only in terms of no of Hyperclique patterns.