Mining weighted sequential patterns using time-interval weight

Tejas R	16CO148
Prajval M	16CO234
Soham Patil	16CO249

Motivation

The purpose of this project is to mine weighted sequential patterns from a Sequence Database (SDB) using the time-interval between the different itemsets of a sequence as a measure of weight of the sequence.

Table 1
A sequence database with a time stamp list.

sid	Sequence	Time stamp list
10	$\langle a, (abc), (ac), d \rangle$	$\langle 0, 1, 2, 3 \rangle$
20	$\langle (ad), c, (bc), (ae) \rangle$	$\langle 1, 2, 3, 4 \rangle$
30	$\langle (ad), (bc), (df) \rangle$	$\langle 1, 3, 5 \rangle$
40	$\langle a, (abc), d \rangle$	$\langle 2, 3, 4 \rangle$

Generalised Vs. TiWS sequence pattern mining

- Generalized sequence pattern mining only takes into account the order of itemsets in a particular sequence.
- Time-interval weighted sequence (TiWS) pattern mining also considers the time intervals between the different itemsets in a particular sequence.
- Lower the time interval between the sequences , higher the importance (weight) given to that sequence.

Algorithm

- Time-interval between pair of itemsets i and j :

$$TI_{ij} = t_j - t_i.$$

Possible pairs of itemsets.

1st Itemset	2nd Itemset	Time-interval
a	(*abc*)	1
a	(*ac*)	2
a	*d*	3
(*abc*)	(*ac*)	1
(*abc*)	*d*	2
(*ac*)	*d*	1

Algorithm contd.

- Weighting functions :

(i) *General scale weighting:* $w_g(TI_{ij}) = \delta^{\frac{TI_{ij}}{u}} = \delta^{\frac{t_j - t_i}{u}}$ *[WF_1].*

(ii) *Log scale weighting:* $w_l(TI_{ij}) = \delta^{\log_2\left(1+\frac{TI_{ij}}{u}\right)} = \delta^{\log_2\left(1+\frac{t_j - t_i}{u}\right)}$ *[WF_2].*

(iii) *General scale weighting with a ceiling:* $w_c(TI_{ij}) = \delta^{\left\lceil\frac{TI_{ij}}{u}\right\rceil} = \delta^{\left\lceil\frac{t_j - t_i}{u}\right\rceil}$ *[WF_3].*

- Strength of a pair of itemsets s_i and s_j :

$$ST_{ij} = length(s_i) \times length(s_j)$$

Algorithm contd.

- Time-interval weight of a sequence :

$$W(S) = \begin{cases} \frac{1}{N}\sum_{i=1}^{l-1}\sum_{j=i+1}^{l}\{w(TI_{ij}) \times ST_{ij}\}, \text{ where } N = \sum_{i=1}^{l-1}\sum_{j=i+1}^{l} ST_{ij} \\ \quad \text{and } w(TI_{ij}) \text{ denotes a weighting function} & (l \geqslant 2) \\ 1 & (l = 1) \end{cases}$$

sid	Sequence weight
10	$\{w(1) \times 11 + w(2) \times 5 + w(3) \times 1\}/17 = 0.863$
20	$\{w(1) \times 8 + w(2) \times 6 + w(3) \times 4\}/16 = 0.832$
30	$\{w(2) \times 8 + w(4) \times 4\}/12 = 0.759$
40	$\{w(1) \times 6 + w(2) \times 1\}/7 = 0.887$

($w_g(TI) = \delta^{TI/u}$, $\delta = 0.9$, and $u = 1$).

Algorithm contd.

- Time-interval support of a sequence X :

$$TiW\text{-}Supp(X) = \frac{\sum_{S:(X \subseteq S) \wedge (S \in SDB)} W(S)}{\sum_{S:S \in SDB} W(S)}.$$

- Time-interval weighted sequential pattern : All the sequences X such that

TiW-supp(X) >= minSupport

Simple support Vs. TiW-support

Change of supports (Simple support vs. TiW-support).

Sequences	Simple support	TiW-support
$a(bc)$	1.000	1.000
aa	0.750	0.773
$a(abc)d$	0.500	0.524
(ad)	0.500	0.476
$(ad)(bc)$	0.500	0.476

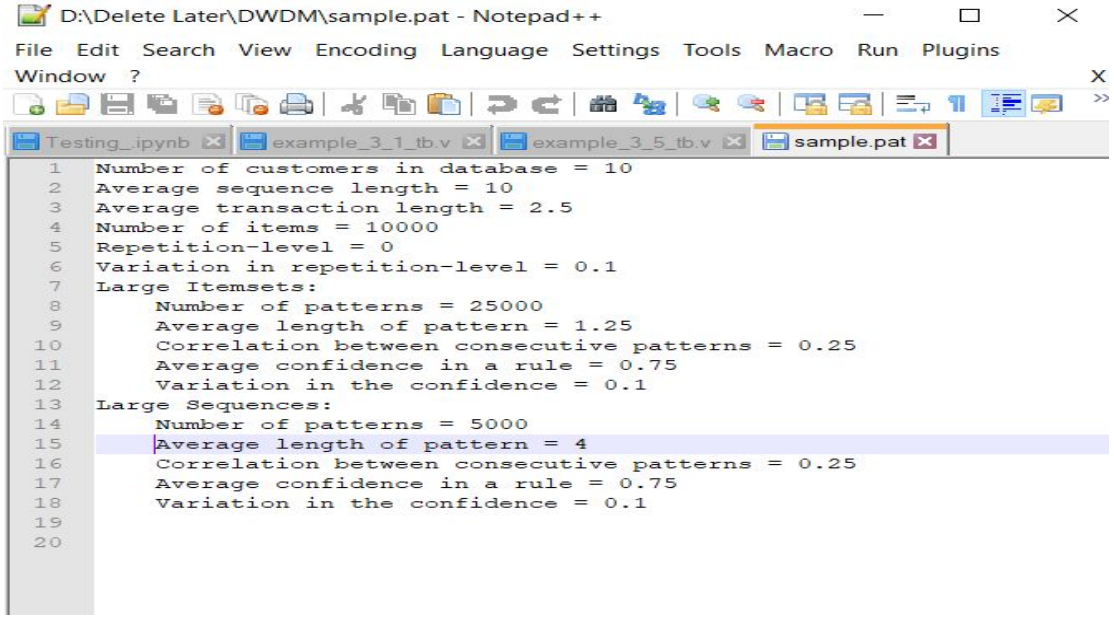
Dataset Generation

An approach using a probability distribution function or that using a randomization function can be considered, but these approaches almost never affect the performance of the new framework of TiWS pattern mining and the proposed psTiWS method.

We are using IBM Synthetic Data Generation Framework for Associations and Sequential Patterns.

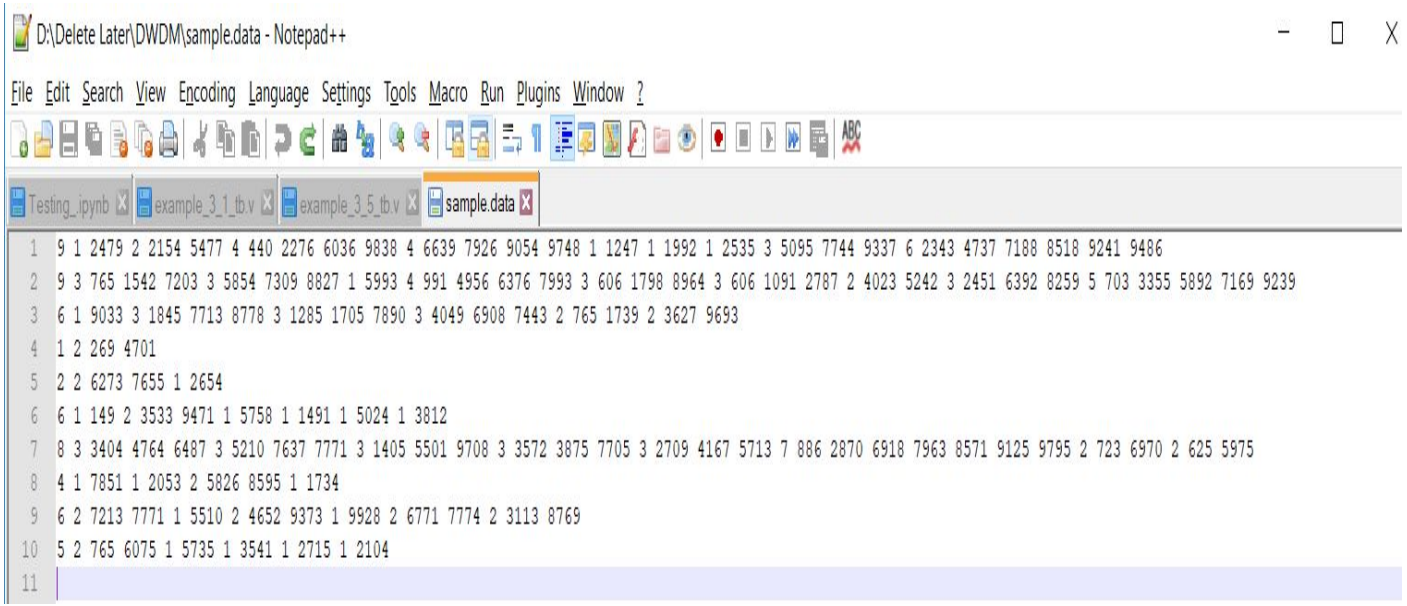
Dataset Generation

Pattern used to generate sample dataset



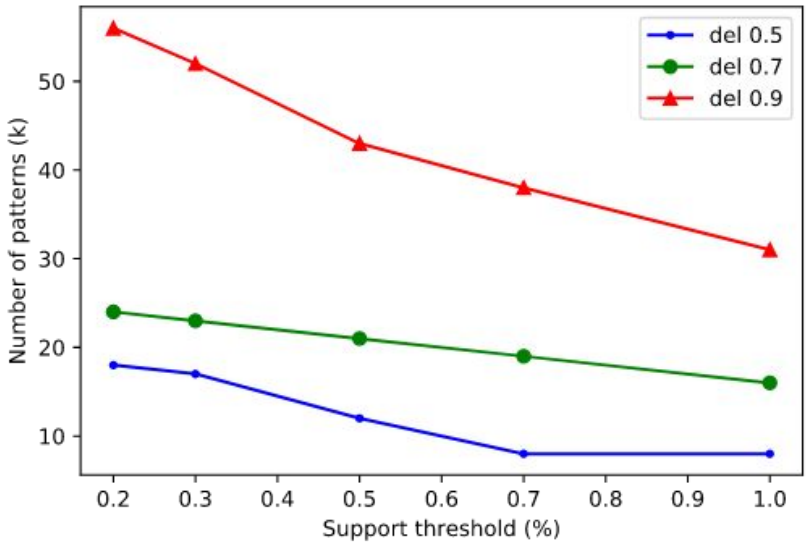
Dataset Generation

Sample dataset generated using IBM framework



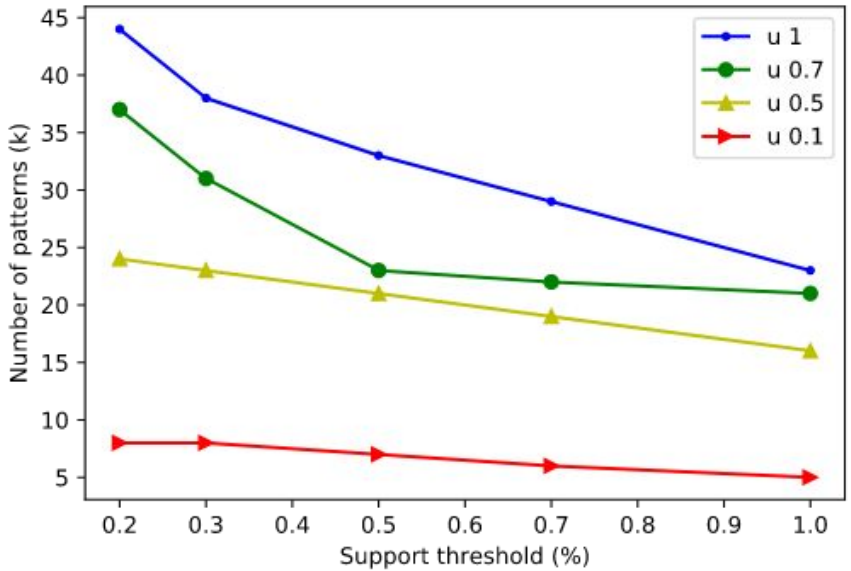
Experimental results

- Number of sequential patterns
  Vs. Support Threshold(%) for various delta values



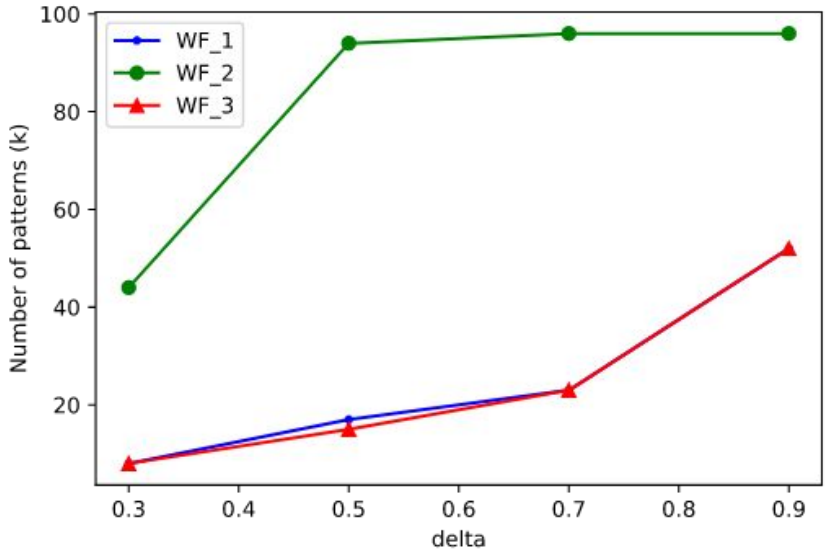
Experimental results contd.

- Number of sequential patterns
  Vs. Support Threshold(%) for various u values



Experimental results contd.

- Number of sequential patterns
  Vs. delta for various weighting functions

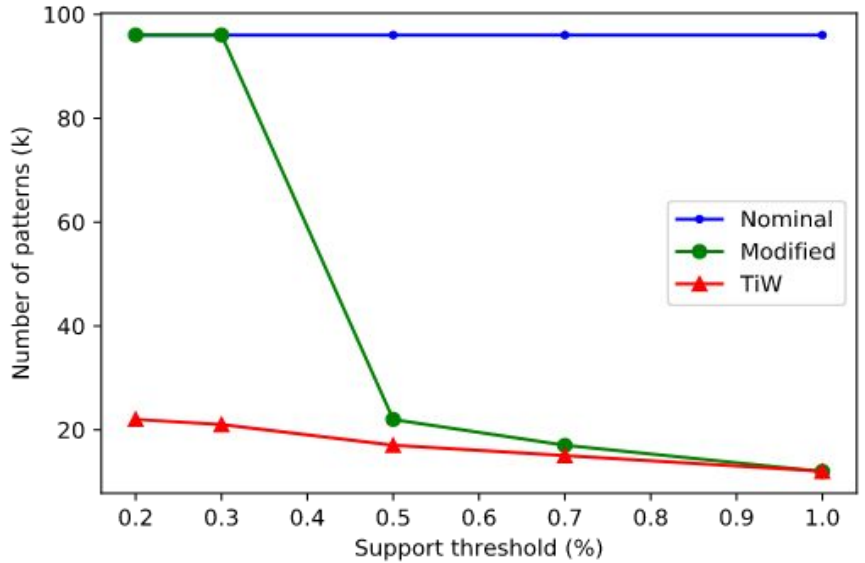


Our contributions

- Modified TiW support for obtaining balance in the number of interesting sequential patterns .
- Making the system more scalable by keeping track of the total TiW supports.
- Greedy algorithm to find the super sequences of a given sequence in database.

Our contributions

- Modified TiW support algorithm with a weightage of 0.7 given to TiW support :



Future Work

- The optimal selection of **u** and **δ** in the time-interval weighting functions .
- Selecting an optimal value of **u** and **δ** based on the characteristics of the application domain and the databases, the proposed approach may be made more effective.
- Modifying WF_2 (log weighted function) to generate frequent patterns and its analysis

Conclusion

- The proposed TiW-support differs from conventional support by considering both the order of sequences and the time-intervals of sequences into account.
- This gives interesting and more valuable sequential patterns .
- In the real world domains , not only the generation order but also the generation times and the generation intervals play a significant role.

Thank You