
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.

<i>sid</i>	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*: $w_g(TI_{ij}) = \delta^{\frac{n_{ij}}{u}} = \delta^{\frac{t_j - t_i}{u}}$ [WF_1].

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

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

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

<i>sid</i>	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	$\{w(1) \times 6 + w(2) \times 1\} / 7 = 0.887$

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

Algorithm contd.

- Time-interval support of a sequence X :

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

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

$$TiW-supp(X) \geq minSupport$$

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

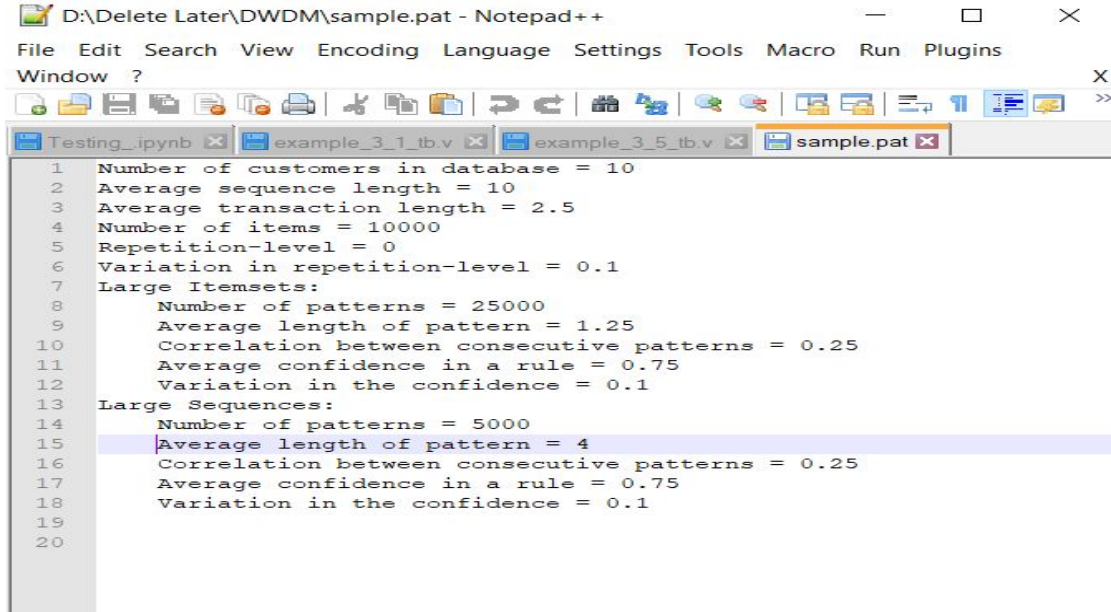
Dataset Generation

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

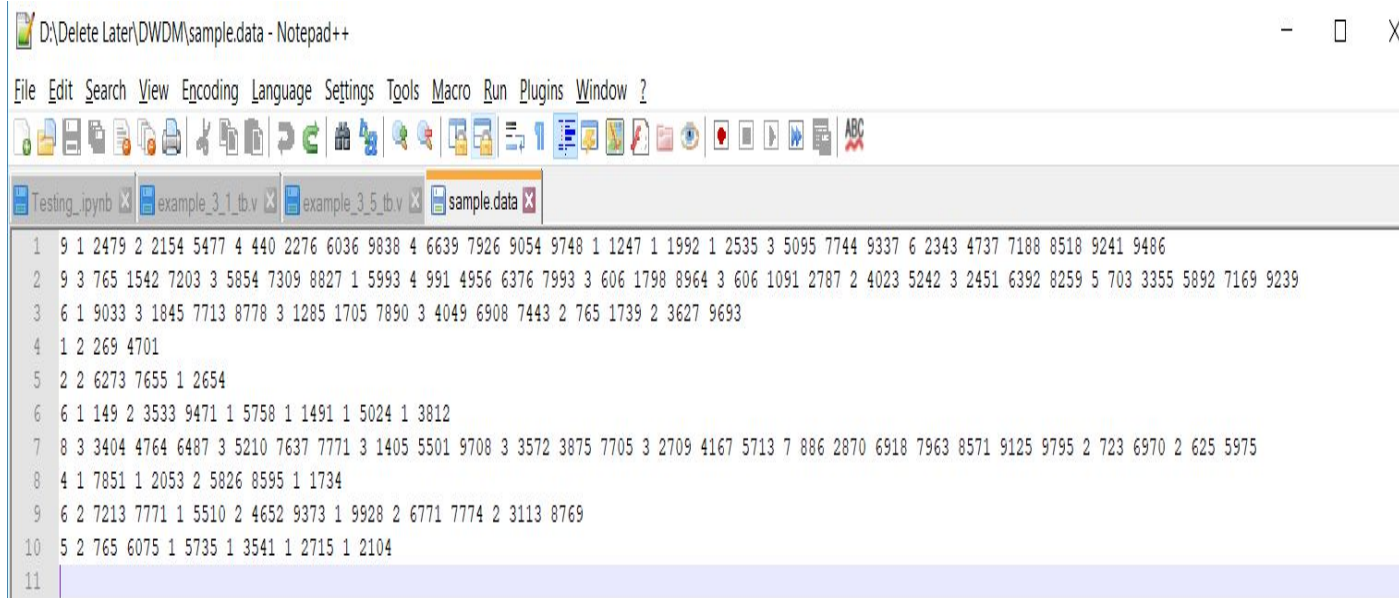
Pattern used to generate sample dataset



```
1 Number of customers in database = 10
2 Average sequence length = 10
3 Average transaction length = 2.5
4 Number of items = 10000
5 Repetition-level = 0
6 Variation in repetition-level = 0.1
7 Large Itemsets:
8     Number of patterns = 25000
9     Average length of pattern = 1.25
10    Correlation between consecutive patterns = 0.25
11    Average confidence in a rule = 0.75
12    Variation in the confidence = 0.1
13 Large Sequences:
14     Number of patterns = 5000
15     Average length of pattern = 4
16     Correlation between consecutive patterns = 0.25
17     Average confidence in a rule = 0.75
18     Variation in the confidence = 0.1
19
20
```

Dataset Generation

Sample dataset generated using IBM framework

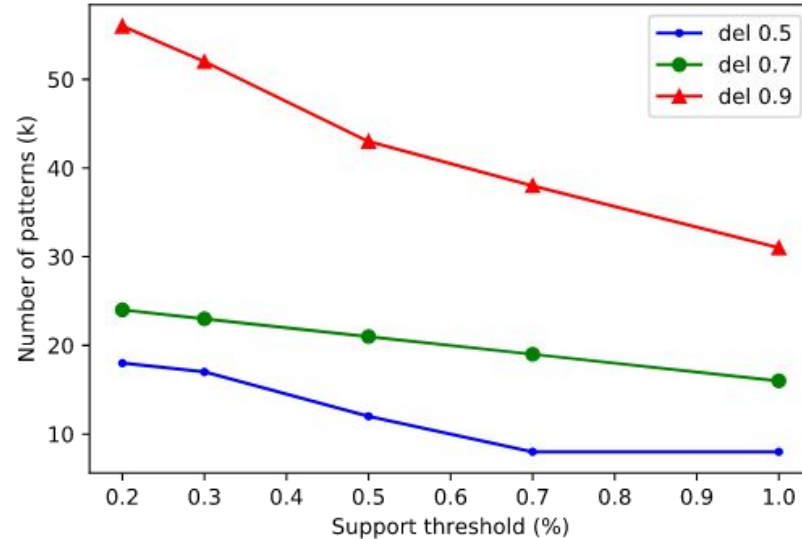


The screenshot shows a Notepad++ window titled "D:\Delete Later\DWDM\sample.data - Notepad++". The window contains a sample dataset with 11 lines of data. Each line consists of a line number followed by a series of integers. The data is as follows:

Line	Data
1	9 1 2479 2 2154 5477 4 440 2276 6036 9838 4 6639 7926 9054 9748 1 1247 1 1992 1 2535 3 5095 7744 9337 6 2343 4737 7188 8518 9241 9486
2	9 3 765 1542 7203 3 5854 7309 8827 1 5993 4 991 4956 6376 7993 3 606 1798 8964 3 606 1091 2787 2 4023 5242 3 2451 6392 8259 5 703 3355 5892 7169 9239
3	6 1 9033 3 1845 7713 8778 3 1285 1705 7890 3 4049 6908 7443 2 765 1739 2 3627 9693
4	1 2 269 4701
5	2 2 6273 7655 1 2654
6	6 1 149 2 3533 9471 1 5758 1 1491 1 5024 1 3812
7	8 3 3404 4764 6487 3 5210 7637 7771 3 1405 5501 9708 3 3572 3875 7705 3 2709 4167 5713 7 886 2870 6918 7963 8571 9125 9795 2 723 6970 2 625 5975
8	4 1 7851 1 2053 2 5826 8595 1 1734
9	6 2 7213 7771 1 5510 2 4652 9373 1 9928 2 6771 7774 2 3113 8769
10	5 2 765 6075 1 5735 1 3541 1 2715 1 2104
11	

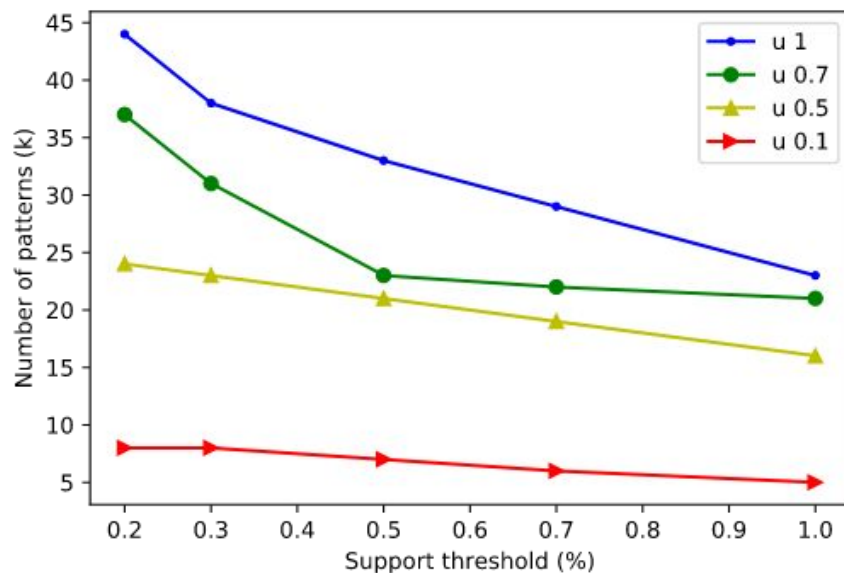
Experimental results

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



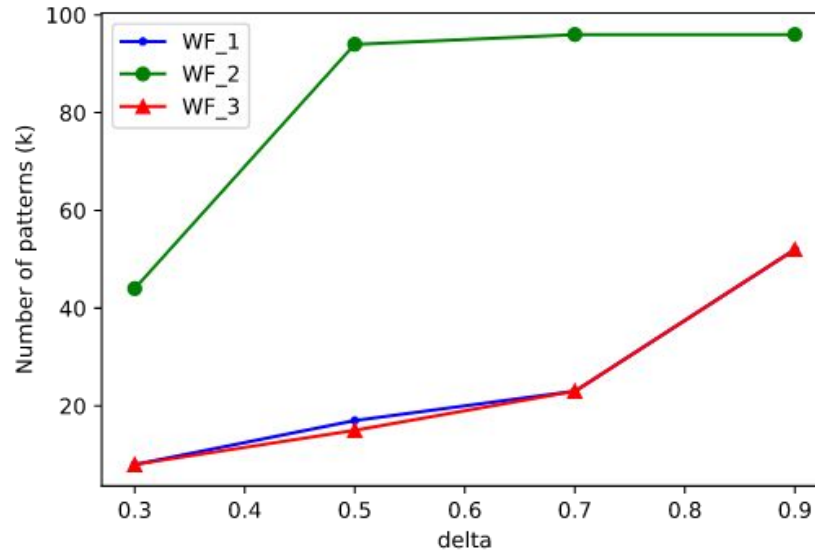
Experimental results contd.

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



Experimental results contd.

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

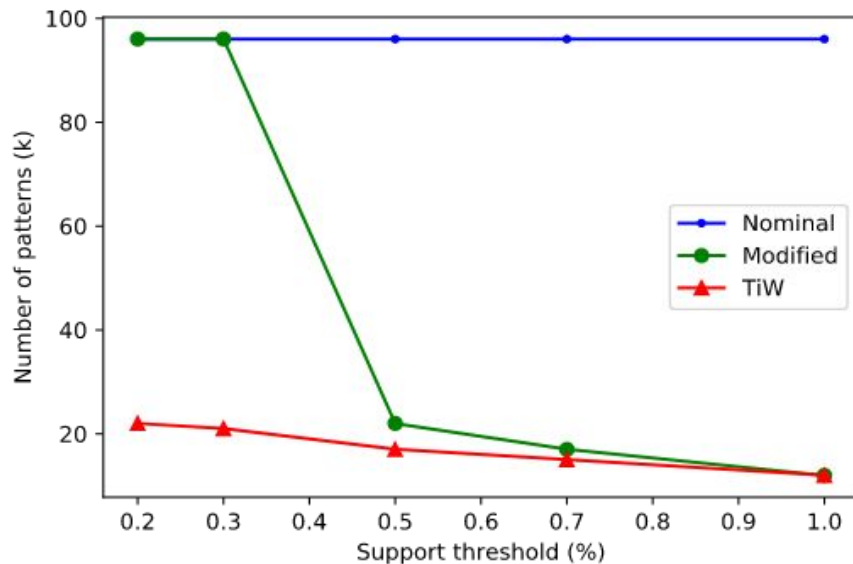


Our contributions

- 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 \mathbf{u} and δ in the time-interval weighting functions .
 - Selecting an optimal value of \mathbf{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
