INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

R&D SHOWCASE 2021
Technology, Social Impact

Discovering Periodic Patterns in Irregular Time Series

Introduction

- Finding (partial) periodic patterns in time series data is a challenging problem of great importance in many applications.
- Most previous studies in this area assumed data as regular time series and disregarded the associated timestamps. But, this assumption is often too restrictive because the data naturally exists as an irregular time series.
- Irregularity is one key characteristic of big data, where irregularity refers to data collection at uneven time stamps. Irregularity within the data is relatively hard to handle due to uneven gaps between the events.
- In this paper, we propose a flexible model of periodic pattern that may be present in irregular time series. Two measures, period and period-support, are employed to determine the interestingness of a pattern in a series.

Model of partial periodic patterns

- Let I be the set of items (or event types). An event e = (t, i), where t ∈ R+ is a timestamp and i ∈ I is an item. A period-segment, PS_i, 1 ≤ i, is an ordered set of events collected within a time interval. That is. $PS_i = \{e_1, e_2, \cdots, e_n\}$, for 1 ≤ i ≤ j ≤ n. A period-segment is said to be empty if there exists no event. An irregular time series S is an ordered collection of periodsegments. That is, $S = PS_1 \cup PS_2 \cup \cdots \cup PS_m$, 1 ≤ m.
- Let $s = \{e_1, e_2, \cdots, e_k\}$, e_p.i ∈ I for 1 ≤ p ≤ k, be a pattern. If a pattern s contains k individual events, then it is a k-pattern. The L-length of s represents the number of unique timestamps in s.
- If $s \subseteq PS_j$, 1 ≤ j ≤ m, it is said that s occurs in PS_j . Let PS^s_j denote the period-segment in which s has occurred in S. Let P $S^s = \{PS^s_p, \cdots, PS^s_q\}$, 1 ≤ p ≤ q ≤ a be the complete set of period-segments in which s occurs.
- Let PS^s_a and PS^s_b , p ≤a ≤ b ≤ q be two consecutive period-segments in PS^s. An inter-arrival time of s, denoted as iat^s = $(PS^s_b - PS^s_a)$.Let $IAT^s = \{iat^s1, \cdots, iat^sx\}$, x = SUP(s)−1, be the complete set of inter-arrival times of s in various period-segments. An $iat^s_k \in IAT^s$, is considered interesting (or periodic) if $iat^s_k \leq maxIAT$, where maxIAT represents the user-specified maximum inter-arrival time
- **Period-Support** : Let $\hat{A} \subseteq IAT^s$ denote the set of all interesting inter-arrival times of s in S. That is, if there exists $iat^s_k \in IAT^s : iat^s_k \leq maxIAT$, then $iat^s_k \in \hat{A}$. The period-support of s represents the number of periodic occurrences of s in the data. That is, period-support(s) = $|\hat{A}|$.
- The frequent pattern s is a **(partial) periodic pattern** if period-support(s) ≥ minPS, where minPS represents the user-defined minimum period-support

Proposed PP-Growth algorithm

Algorithm 1 PP-List (S: time series data, I: set of items, $maxIAT$: maximum inter-arrival time, $minSup$: minimum support and $minPS$: minimum period-support)

```
Let t_l be a temporary array that records the timestamp of the last appearance of each event in S. Let ps_cur denote the number (or position) of current period-segments PS_j.

for each period-segment PS_j ∈ S do
   for each event e ∈ PS_j do
      if e exists in PP-list then
         if (ps_cur − t_l^e) ≤ maxIAT then
            Set ++pf^e.
         end if
         Set ++f^e and t_l^e = ps_cur.
      else
         Add e to the PP-list with f^e = 1, pf^e = 0, and t_l^e = ps_cur.
      end if
   end for
end for
for each event e ∈ PP-list do
   if f^e < minSup or pf^e < minPS then
      Remove e from the PP-list.
   end if
end for
Sort the reperiod-supportning events of the PP-list in descending order of their frequencies.
```

Algorithm 2 PP-Tree (S, PP-list)

```
Create the root node in the PP-tree, Tree, and label it "null".
for j = 1: j ≤ m; ++j do
   Select the periodic events in PS_j and sort them in L order. Let the sorted periodic event list be [e|E], where e is the first event and E is the reperiod-supportning list. Call insert_tree([e|E], j, Tree).
end for
call PP-growth (Tree, null);
```

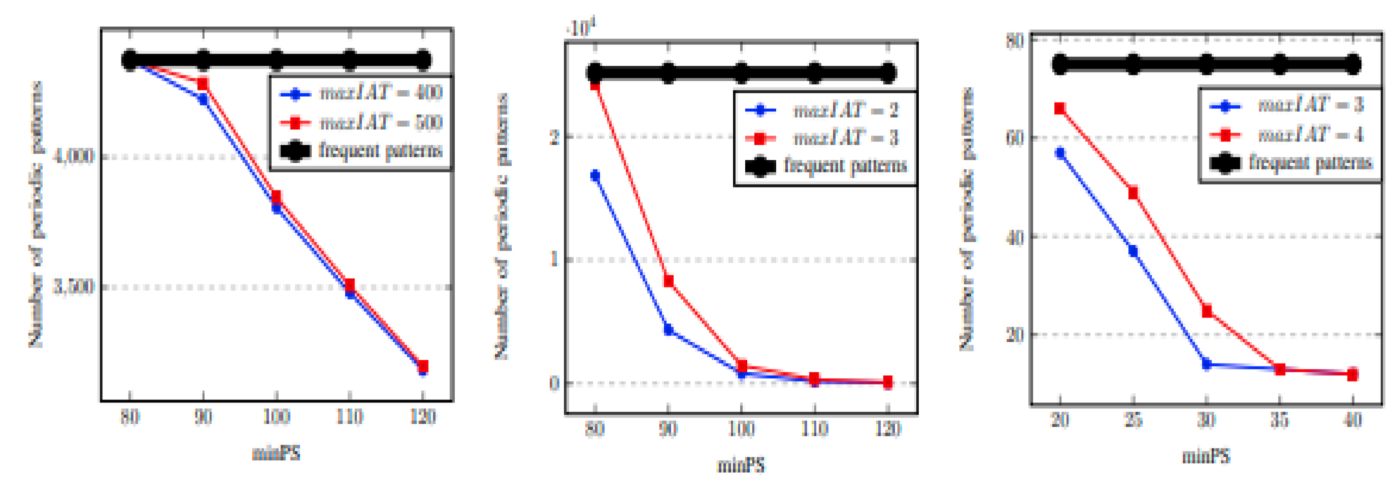
Algorithm 3 insert_tree($[e|E]$, ps_{cur}, $Tree$)

```
while E is non-empty do
   if Tree has a child N such that e.item ≠ N.item and e.timestamp ≠ N.timestamp then
      Create a new node N. Let its parent-node be linked to Tree. Let its node-link be linked to only those nodes that have the same item and timestamp via the node-link structure. Remove e from E.
   end if
end while
Add ps_cur to the leaf node.
```

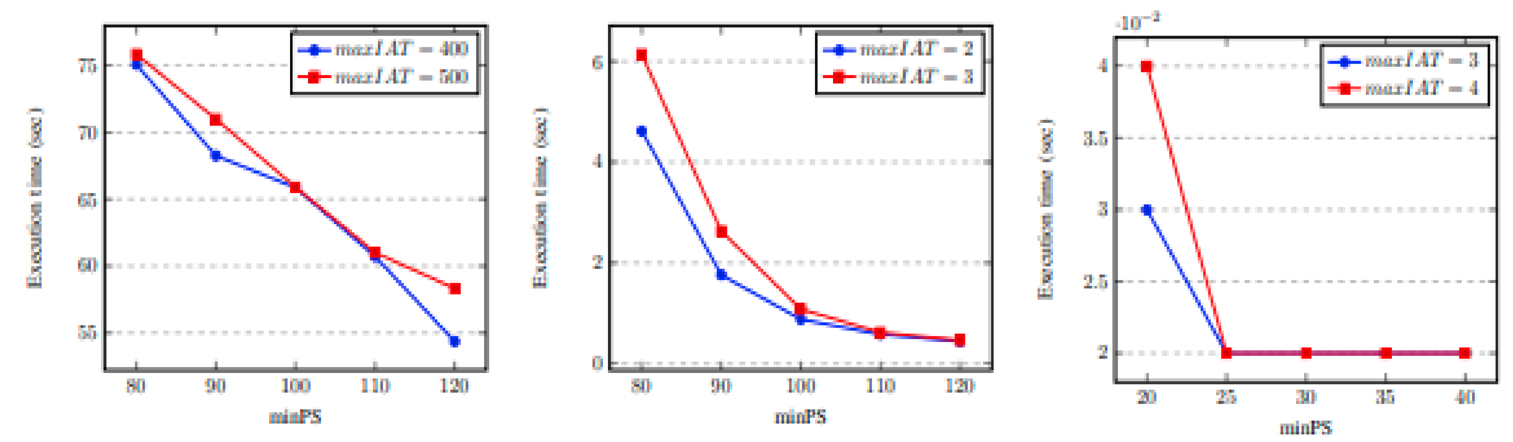
Algorithm 4 PP-growth ($Tree$, α)

```
for each event e_i in the header of Tree do
   Generate pattern β = e_i ∪ α. Traverse Tree using the node-links of β, and construct an array, PS^β, which represents the list of period-segments in which β has appeared in S. Construct β's conditional pattern base and β's conditional PP-tree Tree_β if PS^β.length ≥ minSup and period-support(PS^β) ≥ minPS.
   if Tree_β ≠ ∅ then
      call PP-growth (Tree_β, β);
   end if
   Remove e_i from the Tree and push the e_i's ps-list to its parent nodes.
end for
```

Experiment Results



(a) T10I4D100K (b) Road (c) Patient-1



(a) T10I4D100K (b) Road (c) Patient-1

TABLE III: Some interesting patterns discovered in Road database.

S.No.	Pattern	*support*	*period-support*
1	{(1683, 9:00 hrs), (1683, 10:00 AM), (1181, 16:00 hours), (1181, 17:00PM)},	92	85
2	{(1502,12:00 hrs), (517, 11:00 hrs), (517, 12:00 hrs), (1181, 16:00 hrs), (1181, 17:00 hrs)}	103	102
3	{(1502, 11:00 hrs), (1502, 12:00 hrs), (517, 12:00 hrs), (1473, 20:00 hrs),(1473, 22:00 hrs)}	90	87

Publication

C. Saideep, R. Uday Kiran, K. Zettsu, P. Fournier-Viger, M. Kitsuregawa and P. Krishna Reddy, "Discovering Periodic Patterns in Irregular Time Series," 2019 International Conference on Data Mining Workshops (ICDMW), Beijing, China, 2019, pp. 1020-1028.

Authors Name : C. Saideep,R. Uday kiran, P. Krishna Reddy Research Center: DSAC