Online Frequent Episode Mining

Presented by: Shahab Helmi

Spring 2017

BIG Data Management and Mining Laboratory
UNIVERSITY OF COLORADO DENVER | ANSCHUTZ MEDICAL CAMPUS

Paper Information

Authors:

Xiang Ao [#†1], Ping Luo [#2], Chengkai Li [*3], Fuzhen Zhuang [#1], Qing He [#1]

[#] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, China
[1] {aox,zhuangfz,heq}@ics.ict.ac.cn [2] luop@ict.ac.cn

[*] *University of Texas at Arlington, USA*
[3] cli@uta.edu

[†] *University of Chinese Academy of Sciences, Beijing, China*

Publication:

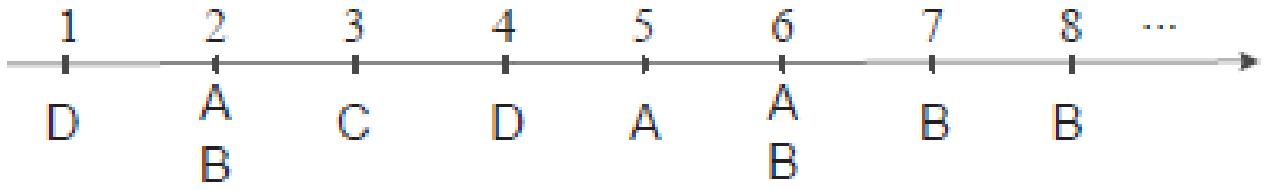
- ICDE 2015

Type:

- Research Paper

Introduction

- Frequent episode mining (FEM): a popular framework for discovering sequential patterns from sequential data.
- Applications: telecommunication, manufacturing, finance, biology, system log analysis, news analysis.
- Here: episode is a sequence of totally ordered (serial) events



Applications for Online FEM

1. HFT (High Frequency Trading)
   - 22 seconds vs hours and days!
2. Predictive maintenance of data centers
   - Avoiding a major crash

- Both will be useful IF predictive models exist (no details provided)

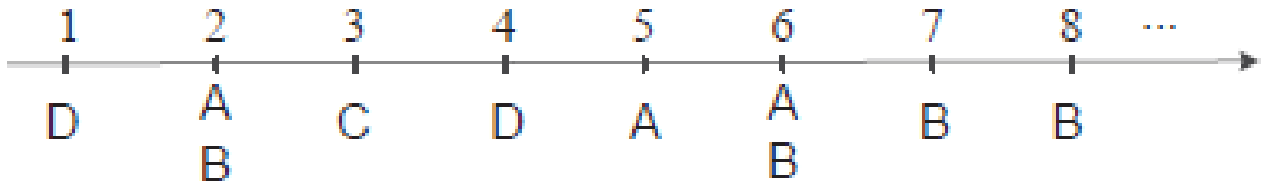
Challenges of Online FEM & Contributions

1. All patterns must be stored:
   - The number of minimal-occurrences is considered as the frequency measure, which is not monotonic
   - Also, dataset is dynamic; hence, frequent patterns may become infrequent and vice versa
2. The tree structure is complicated: temporal information must be stored as well
   - Compact data structure is required -> e-trie
3. Recency effect: only freshest pattern are of interest
   - User can set the expiration time
4. Time critical analysis
   - Heuristic: last episode occurrence

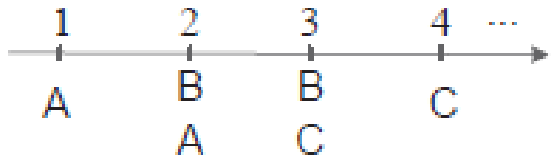
Related Work

- Offline frequent episode mining
  - BFS vs DFS
  - Different domains
  - Different frequency measures
- Probabilistic frequent episode mining –> events are uncertain
- High-utility episode mining: each event has a weight –> non-monotonic
- Online frequent itemset mining

Definitions



- Episode α: $D -> A -> C$ is a 3-episode
- Sub-episode and super episode
- Occurrence and Minimal-occurrence
  - Non-monotonic:
    - $|MoSet(A -> B -> C)| = 2$
    - $|MoSet(A -> C)| = 1$



- Frequent episodes: $if\ |MoSet(\alpha)| \geq \min\_sup, \delta$

Problem Definition

- ***Batch mode***: *Given an event sequence S, a minimum support threshold min_sup and a maximum occurrence window threshold δ, the frequent episode mining problem is to find all frequent episodes in S.*

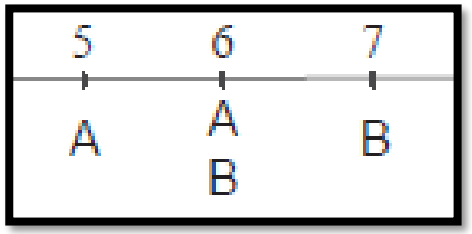
- ***Online mode***: *Given S, a min_sup, a δ, a Δ, the frequent episode mining problem is to find all frequent episodes in the last Δ time stamps.*
  - Usually $\Delta \gg \delta$

Solution Overview

$min\_\sup = 2, \delta = 4, \Delta = 7$

- M_i^j all minimal episode occurrences in $[t_i, t_j]$

For example, in Figure 1, $M_5^7 = \{(\mathsf{A}, [5,5]), (\mathsf{A} \rightarrow \mathsf{A}, [5,6]), (\mathsf{A} \rightarrow \mathsf{B}, [5,6]), (\mathsf{A} \rightarrow \mathsf{B} \rightarrow \mathsf{B}, [5,7]), (\mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{B}, [5,7])\}$.



Theorem 1: Given a sequence $\vec{S}$ with the time stamps starting from 1 to k, $\mathcal{M} = \{M_1^\delta \cup M_2^{\delta+1} \cup \cdots \cup M_{k-\delta}^{k-1} \cup M_{k+1-\delta}^k \cup M_{k+2-\delta}^k \cup \cdots M_{k-1}^k \cup M_k^k\}$ contains all the minimal episode occurrences in this sequence.

Solution Overview (cont'd)

- M can be divided into two disjoint sets:

$$\mathcal{M}_{ex} = \{M_1^{\delta}, M_2^{\delta+1}, \cdots, M_{k-\delta}^{k-1}, M_{k-\delta+1}^{k}\}$$

$$\mathcal{M}_{in} = \{M_{k-\delta+2}^{k}, M_{k-\delta+3}^{k}, \cdots, M_{k-1}^{k}, M_{k}^{k}\}$$

- When new data arrives, only episodes in M_{in} are affected so M_{ex} can be stored in an external storage

Solution Overview (cont'd)

1. Add the new time stamp:

$$\mathcal{M}_{in} \leftarrow \mathcal{M}_{in} \cup \{M_{k+1}^{k+1}\}$$

2. Update the upper indexes

$$\mathcal{M}_{in}^{'} = \{M_{k-\delta+2}^{k+1}, M_{k-\delta+3}^{k+1}, \cdots, M_{k}^{k+1}, M_{k+1}^{k+1}\}$$

3. Move the first one to M_{ex}

$$\mathcal{M}_{in} \leftarrow \mathcal{M}_{in}^{'} - \{M_{k-\delta+2}^{k+1}\}$$

$$\mathcal{M}_{ex} \leftarrow \mathcal{M}_{ex} \cup \{M_{k-\delta+2}^{k+1}\}$$

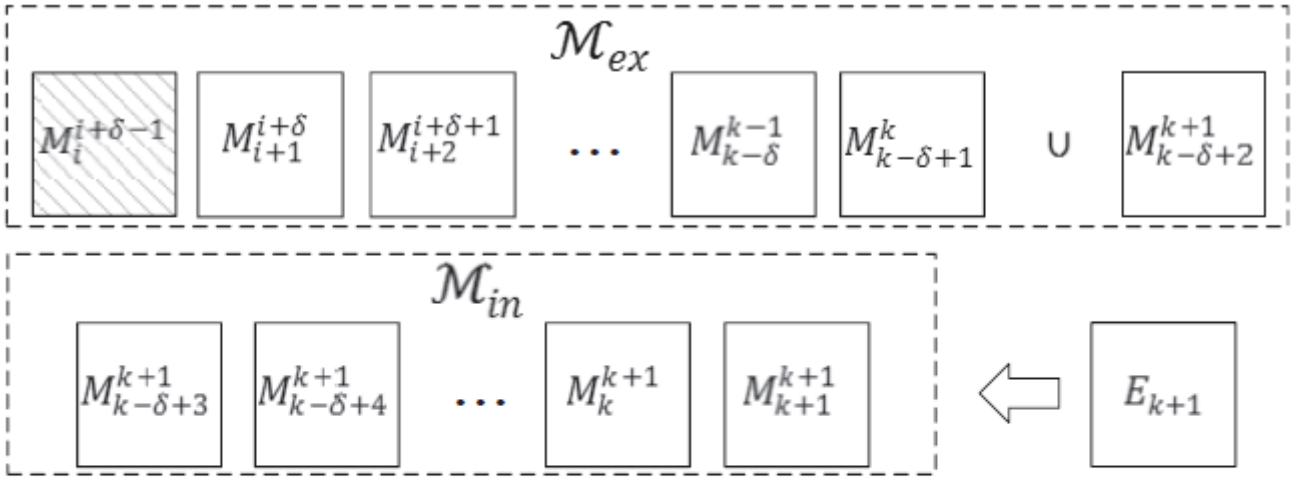
Solution Overview (cont'd)

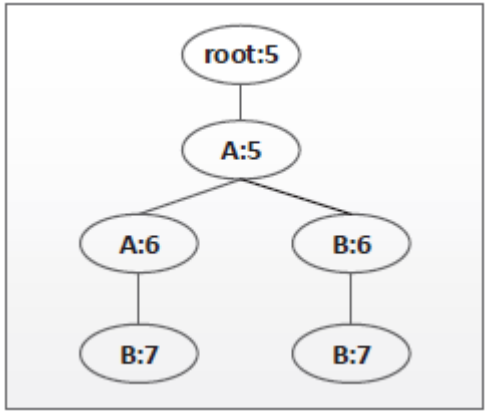
- If $\Delta == \infty$:
  1. Update frequency counts
  2. Update the set of frequent and infrequent sets

- Else
  1. Delete expired occurrences
  2. Update frequency counts
  3. Update the set of frequent and infrequent sets

Solution Overview (Storage Framework)

- The first scan is not possible –> sharp increase in memory usage





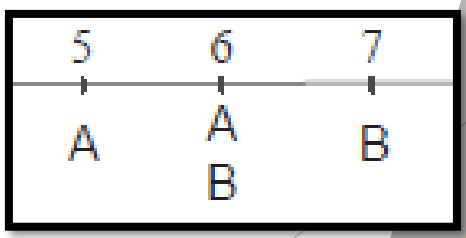
The episode trie $\mathcal{T}_5^7$.

Frequent Episode Set

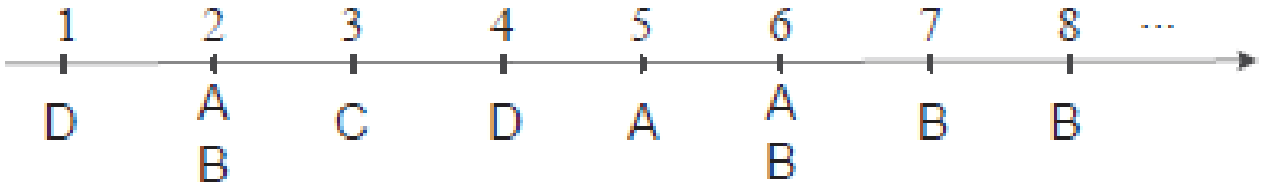
Key	Value
α_j	$\mathrm{sp}(\alpha_j)$
...	...

Infrequent Episode Set

Key	Value
α_i	$\mathrm{sp}(\alpha_i)$
...	...



The Last Episode Occurrence



See the running example in Figure 1. Consider the time window of $[4, 7]$. (A→B, $[6, 7]$) is the last occurrence of episode A→B within this time window. However, (A→B, $[5, 6]$) is not the last occurrence because of the existence of (A→B, $[6, 7]$).

- Then each M_i^j (and their corresponding trie) can be divided into two disjoints sets:

$$M_j^k = \begin{cases} S_j^k = M_j^k \cap L_{k-\delta+1}^k \\ \overline{S_j^k} = M_j^k - (M_j^k \cap L_{k-\delta+1}^k) \end{cases}$$

Complexity Analysis

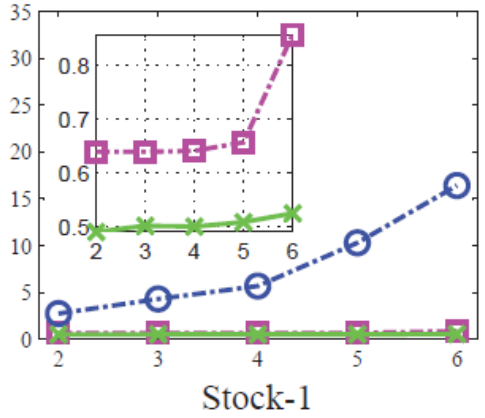
- Both space and time complexities are $O(m^{\delta})$ where $\mathrm{m} = |\Sigma|$

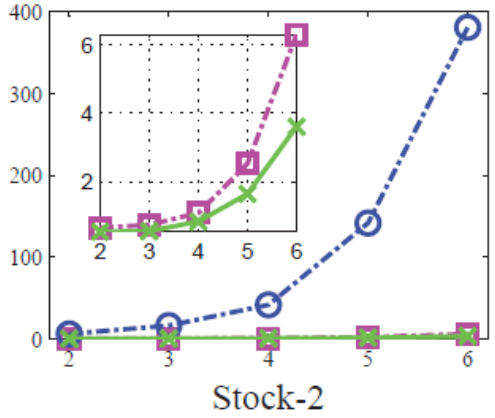
Experiments: Datasets

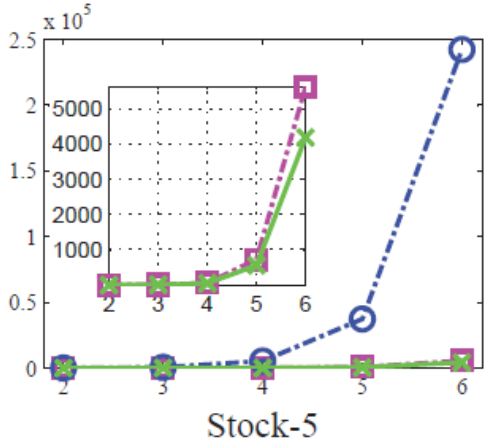
- Stock: from china stock market, each one is for a different industry
  - Discretized
- Kosarak and BMS: click stream data
- chinaStore and Retail: basket data

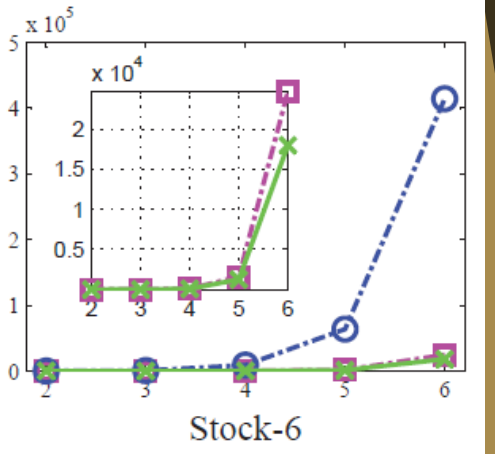
Data set	#Time stamp	#Events	Avg. #Events per Time stamp
Stock-1	2509	4	1.0
Stock-2		8	2.4
Stock-3		16	4.7
Stock-4		24	7.0
Stock-5		32	9.5
Stock-6		40	11.4
Retail	88,162	16,470	10.3
Kosarak	990,002	41,270	8.1
chainStore	1,112,949	46,086	7.3
BMS	59,601	497	2.5

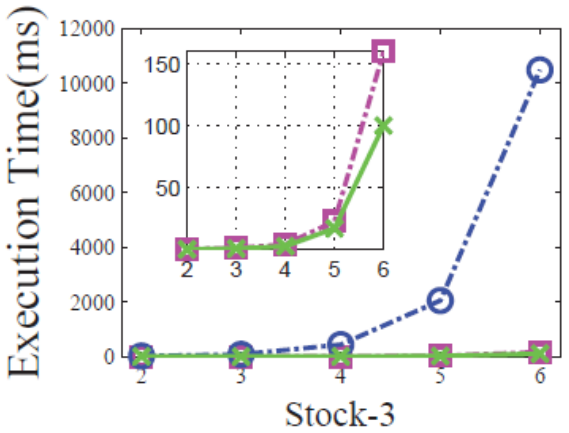
Experiments: Online Mode

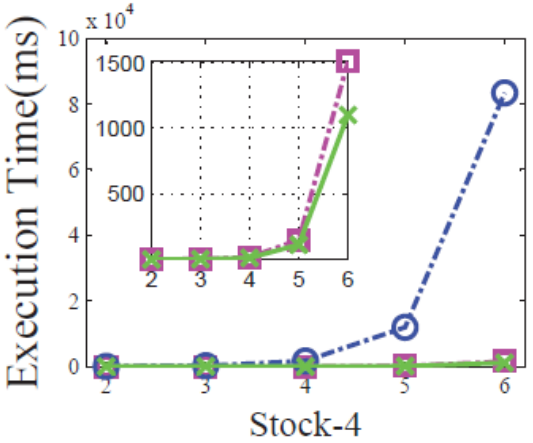












MESELO-BS Brute MESELO

Experiments: Batch Mode