Sequence Prediction using Partially-Ordered Episode Rules

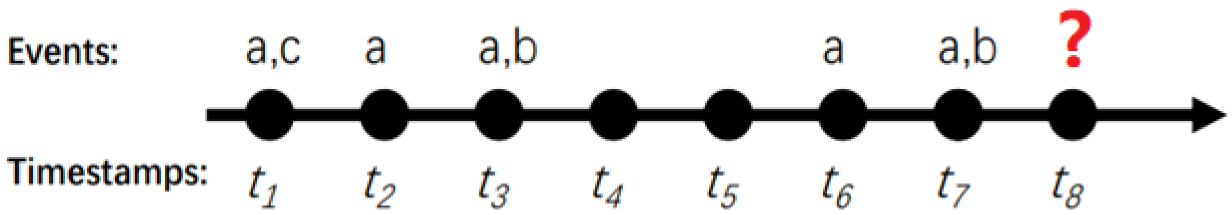
Yangming Chen[1], Philippe Fournier-Viger[1,2], Farid Nouioua[2], Youxi Wu[3]

1 Harbin Institute of Technology (Shenzhen), China
2 Shenzhen University, China
3 University of Bordj Bou Arreridj, Algeria
4 Hebei University of Technology, Tianjin, China

UDML 2021 @ ICDM 2021

Introduction

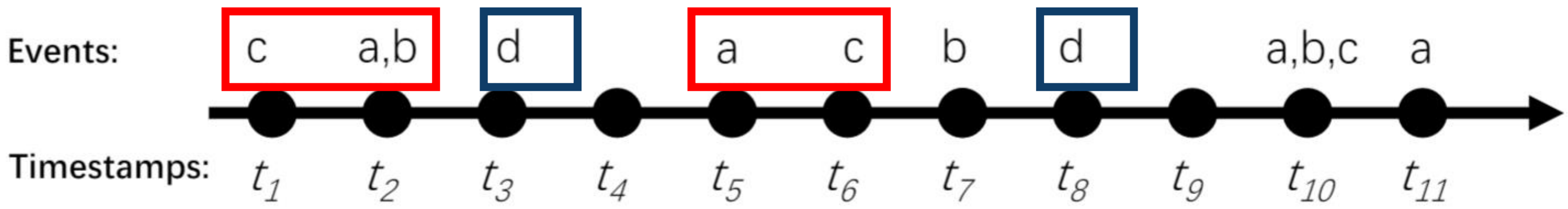
- **Sequence prediction**:
  predicting the next event/symbol of a sequence



- **Episode rule mining**: discovering rules to help understand the data or do predictions.
- We compare **three types of rules** for prediction:
  - **Episode rules** (Toivonen et al., 1996)
  - **Precise positioning episode rules** (Ao et al., 2017)
  - **Partially-ordered episode rules** (Fournier-Viger et al., 2021)

Three Types of Episode Rules

1) Standard episode rule (SER): rule where events in the antecedent and in the consequent are ordered (found by MINEPI+ algorithm).



Examples:

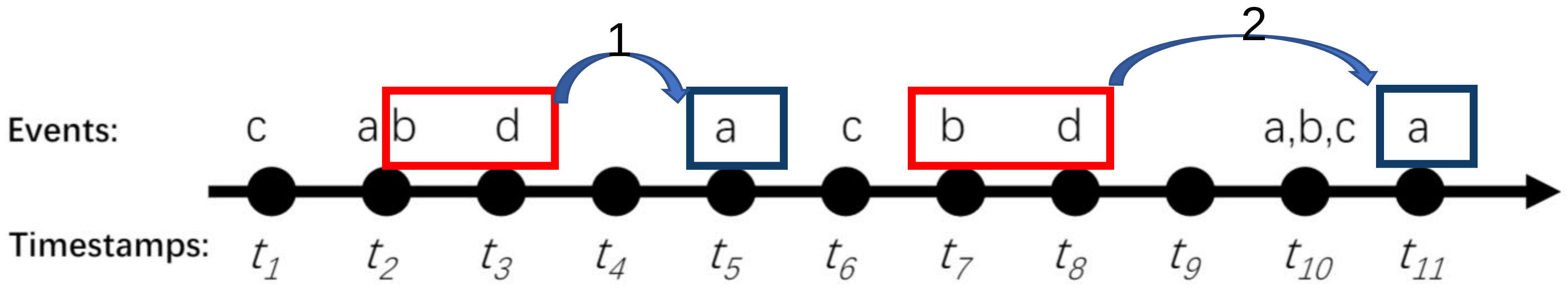
The rule <{c}, {a}> → <{d}> has an occurrence in time interval [t1, t3]

The rule <{a}, {c}> → <{d}> has an occurrence in time interval [t5, t8]

Support: How many times a rule appears
Confidence: The conditional probability that the antecedent is followed by the consequent

Three Types of Episode Rules

2) Precise-Positioning Episode rule (PER): rule where the elapsed time between the antecedent and the consequent is fixed (found by PRU algorithm).



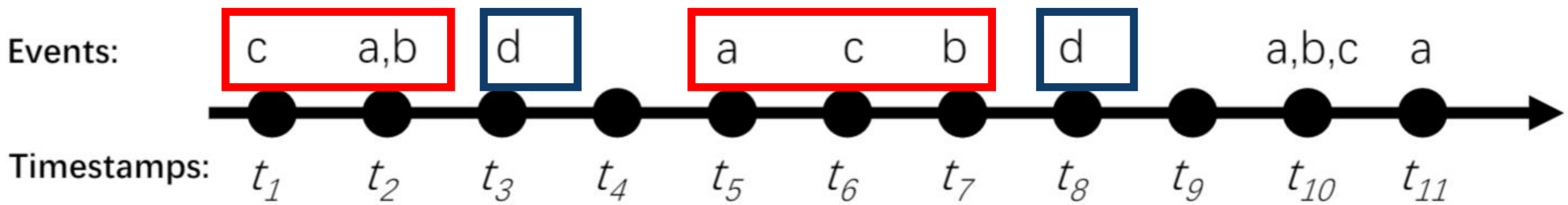
Examples:

The rule <{b}, {d}> $\xrightarrow{1}$ <{a}> has an occurrence in time interval [t2, t5]

The rule <{b}, {d}> $\xrightarrow{2}$ <{a}> has an occurrence in time interval [t7, t11]

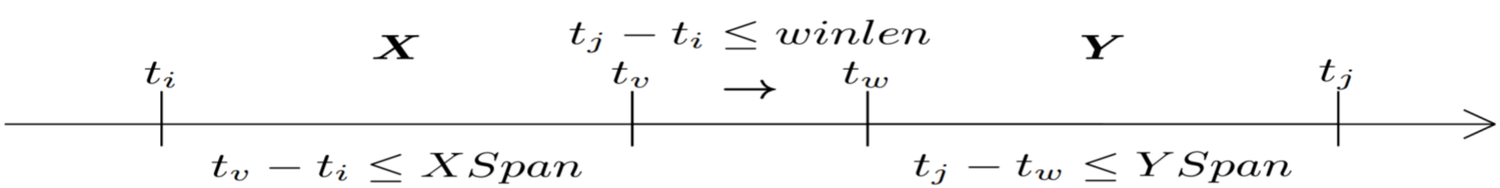
Three Types of Episode Rules

3) Partially-Ordered Episode Rules (POER): rule where events in the antecedent and in the consequent are unordered (found by POERM algorithm).



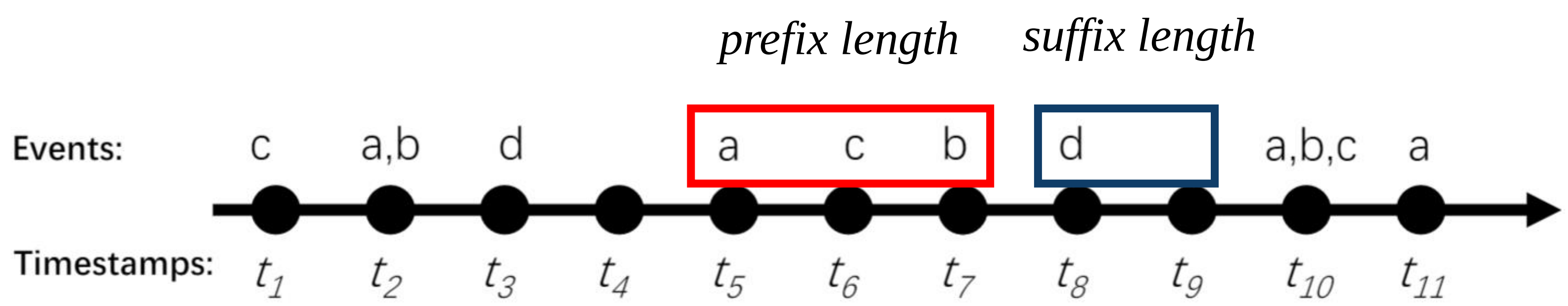
Example:
The rule <{a,b,c}> → <{d}> has occurrences in [t1, t3] and [t5, t8].

minsup= 3, minconf= 0.6, XSpan= 3, winlen= 4, YSpan= 1



Sequence Prediction

Goal: predict an event that will appear in the *suffix* of a sequence given the events observed in the *prefix.*

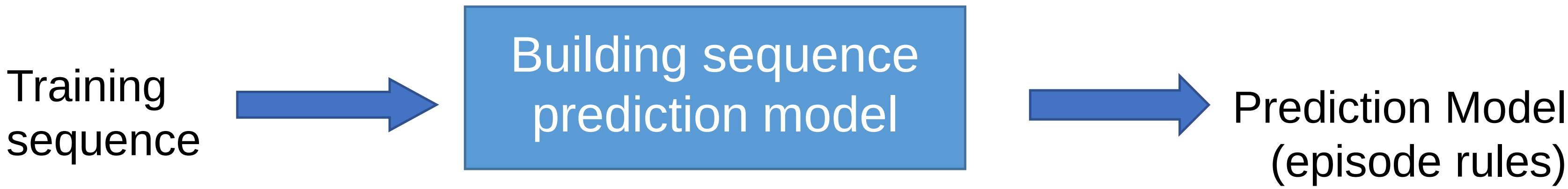


Possible outcomes:

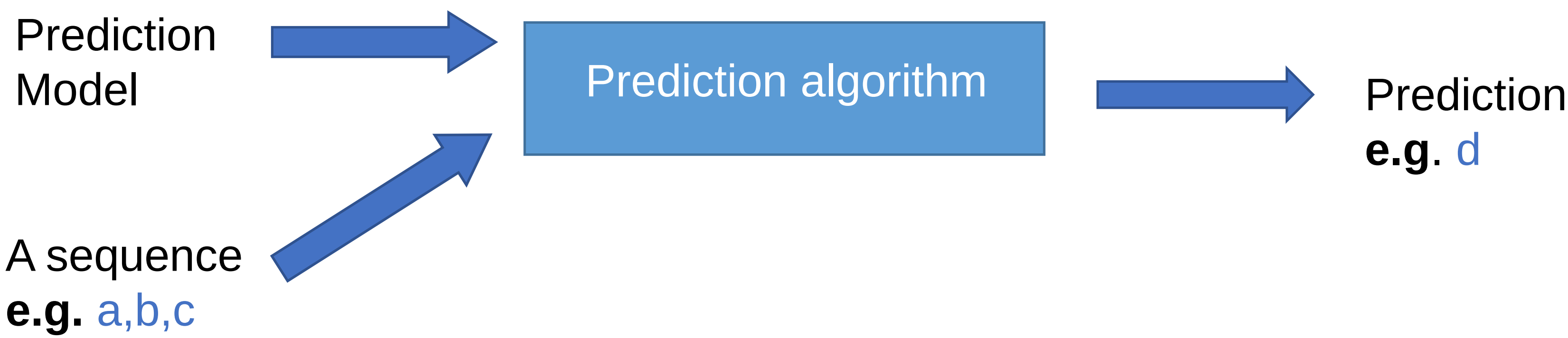
- Good prediction
- Wrong prediction
- Unable to make a prediction (no match)

3. The EpisodePredictor Framework

Phase 1) Training



Phase 2) Prediction



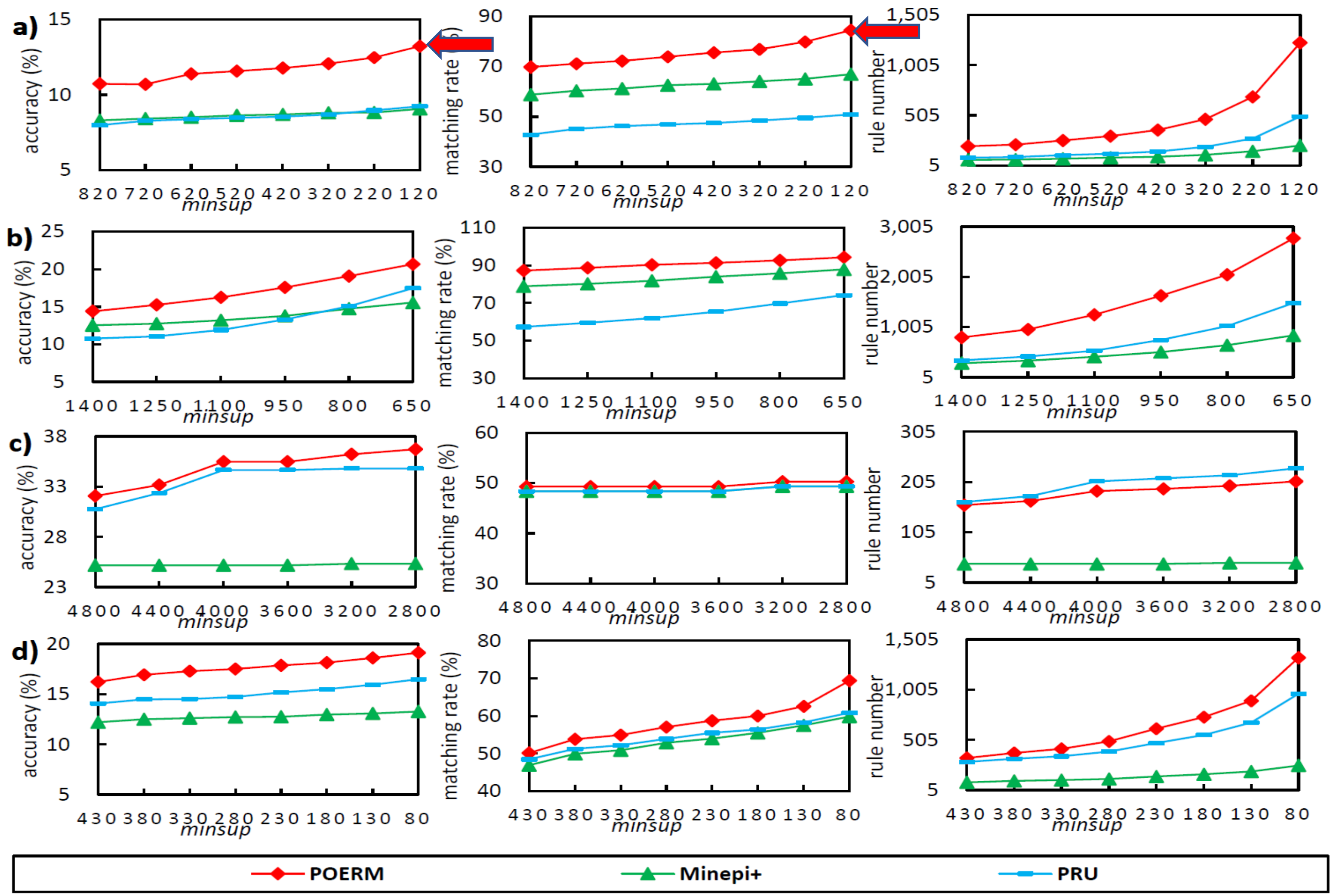
4 Experimental Evaluation

Dataset and Default Values

Dataset	# Timestamps	# Events
Bible	649,024	13,905
OnlineRetail	2,364,798	2,603
FIFA	710,435	2,990
Leviathan	153,682	9025

Default Values	minsup	minconf	prefixSize	suffixSize
Bible	120	0.3	4	1
OnlineRetail	650	0.3	4	1
FIFA	2800	0.3	2	2
Leviathan	80	0.3	2	2

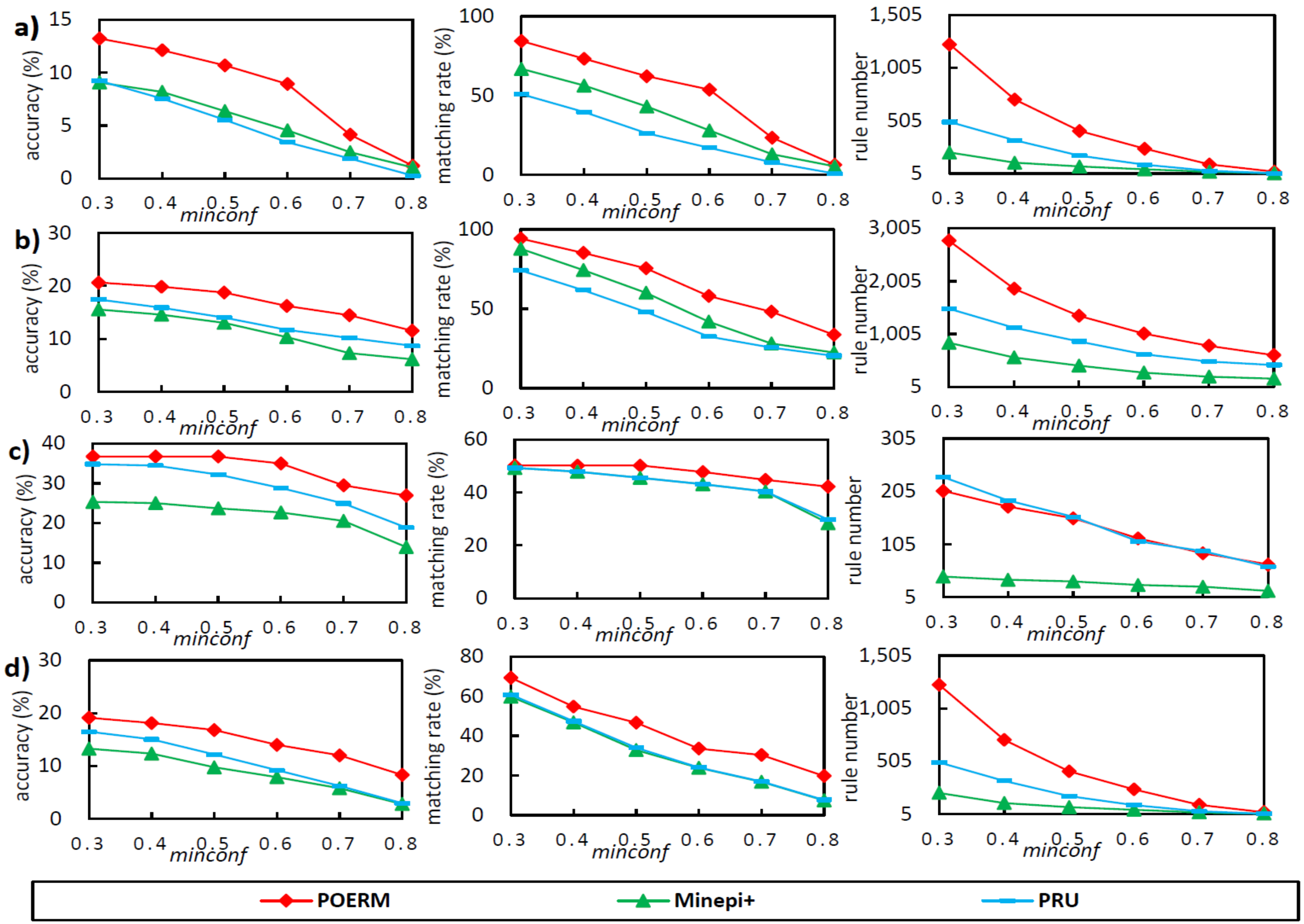
4. Experimental Evaluation (Influence of *minsup*)



Generally, $\uparrow minsup \Rightarrow \uparrow rule\ count, \uparrow matching\ rate, \uparrow accuracy$

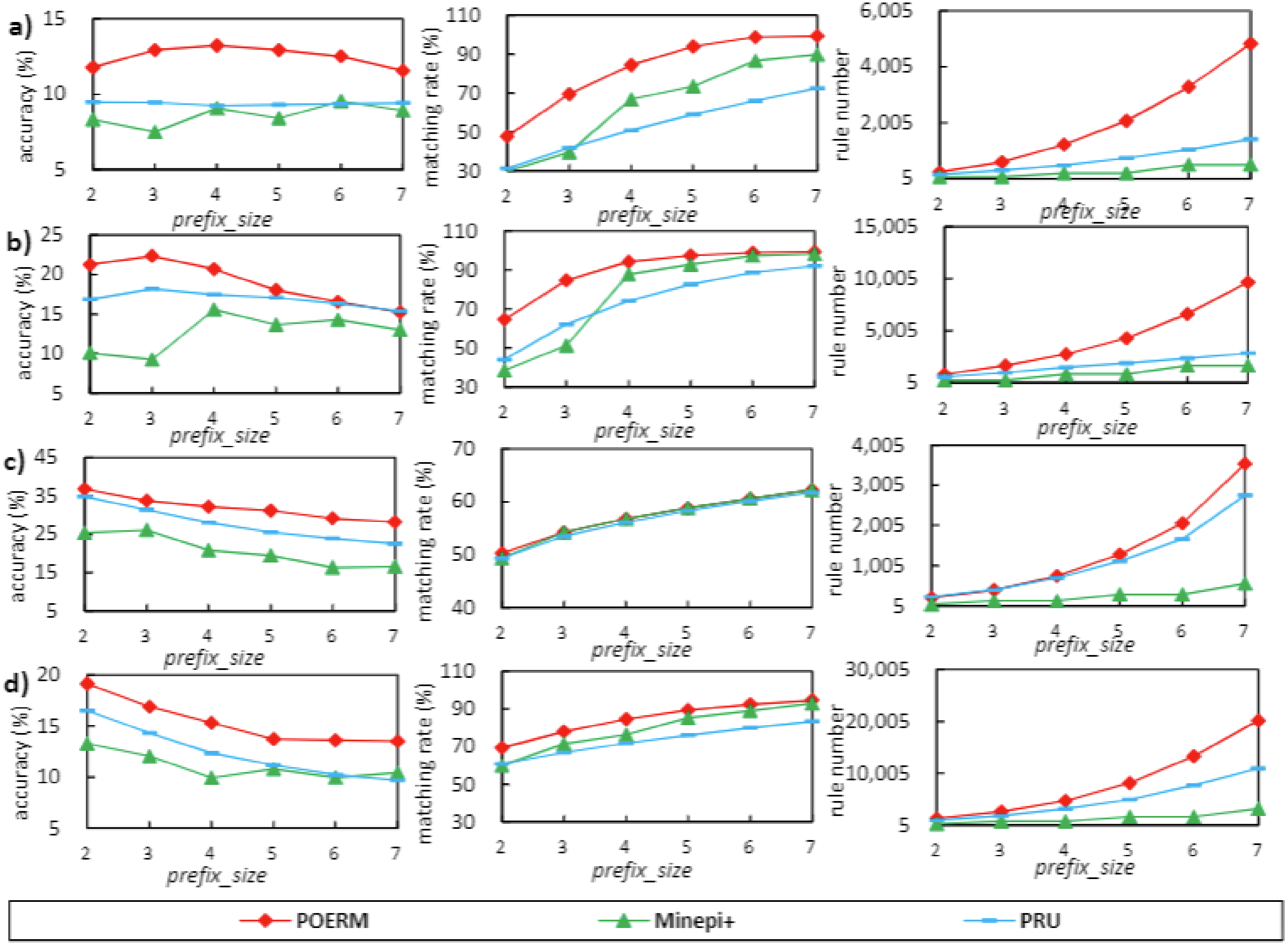
Partially-ordered episode rules have the best results.

4. Experimental Evaluation (Influence of *minconf*)



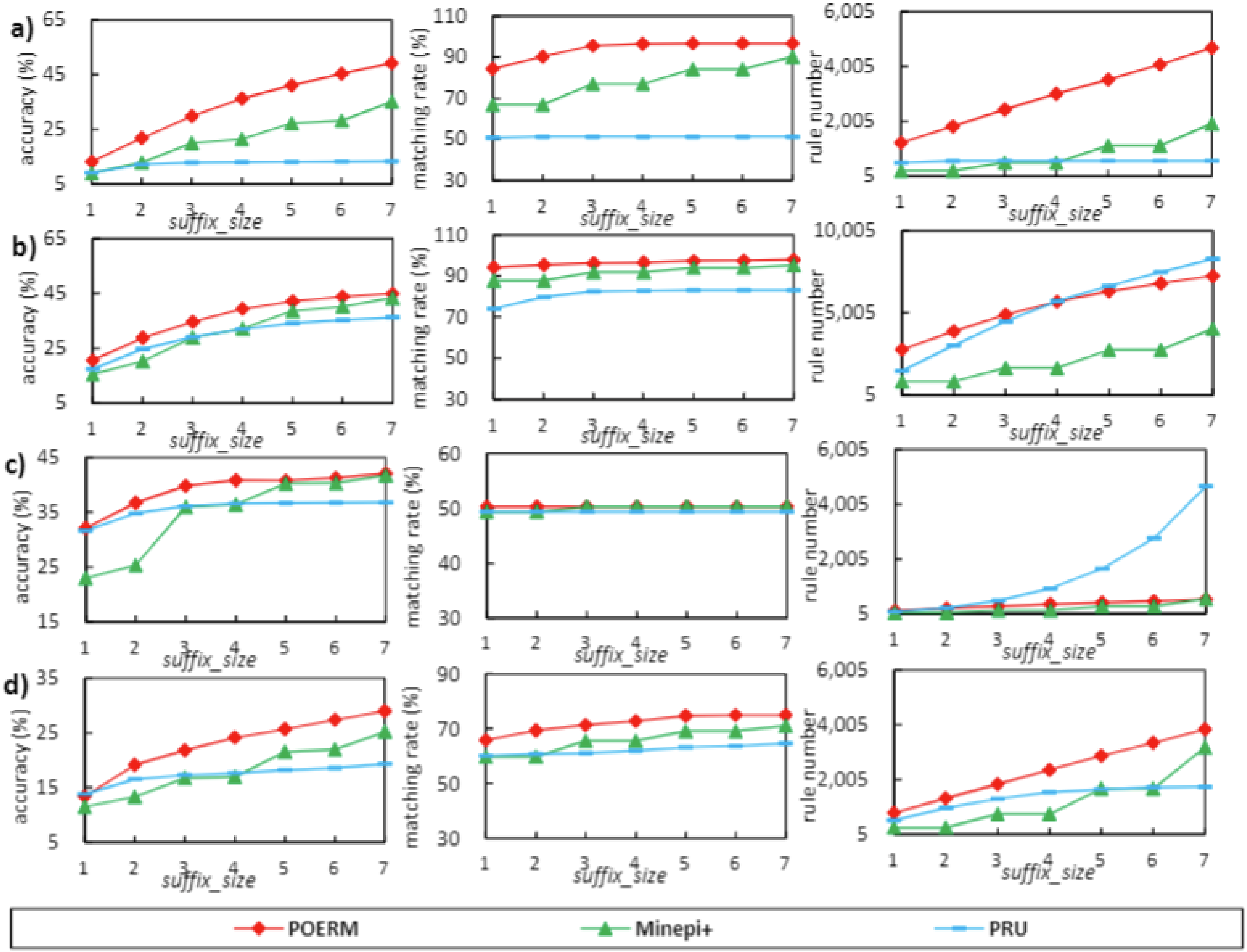
Generally, $\uparrow minconf \Rightarrow \downarrow rule\ count, \downarrow matching\ rate, \downarrow accuracy$

4. Experimental Evaluation (Influence of *prefix size*)



Generally, ↑ $prefix\ size\ \Rightarrow$↑ $rule\ count$,↑ $matching\ rate$,↑↓ $accuracy$

4. Experimental Evaluation (Influence of *suffix* size)



Generally, $\uparrow suffix\ size \Rightarrow \uparrow rule\ count, \uparrow matching\ rate, \uparrow\ accuracy$

5. Conclusion

- **Contributions:**
  - ➢utilize a new type of episode rules, named **partially-ordered episode rules to improve** sequence prediction
  - ➢Better results are obtained in terms of **accuracy and matching** rate compared to two popular types of episode rules
- **Future work:**
  - ➢Episode rule-based sequence prediction models for more complex event sequences.
  - ➢Design an incremental episode rule mining algorithm to decrease runtime performance in dynamic environment

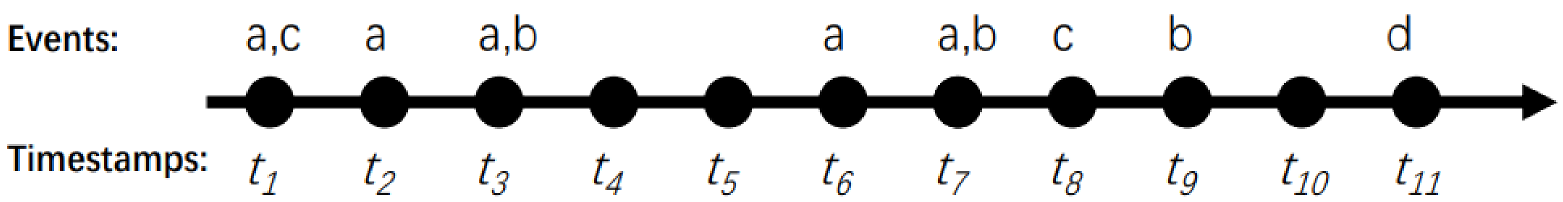
➢**Open-source code, datasets:**

- ➢SPMF data mining library (over 200 algorithms)

Thanks for listening!

Frequent Episode Mining

Input: a sequence of events with timestamps



Output:

All frequent episode
(*minimal occurrence* $\geq$ *minSup*)

For example:

episode <{a,b},{c}>

Episode types:

parallel episodes, serial episodes, complex episodes

Algorithms:

WINEPI, MINEPI, EMMA, MINEPI+, ...