Mining Closed Discriminative Dyadic Sequential Patterns

David Lo[1], Hong Cheng[2], and Lucia[1]

[1]Singapore Management University
[2]Chinese University of Hong Kong

Motivation: Sequence Pairs

- Much data is in sequential formats
  - Sequence of words in a document
  - Nucleotides in a DNA
  - Program events in a trace, etc
- Focus: sequence pairs
  - Each data unit is composed of 2 sequences
  - Each data unit is given a label: +ve or –ve
- Mine discriminative patterns that distinguishes +ve pairs from –ve pairs

Motivation: Sequence Pairs

- NLP: Language translation
  - Original-translated text = pair of sequences of tokens
  - Label: Good vs. bad translations
- Software Engineering: Duplicate bug reports
  - Users report bugs in an uncoordinated fashion
  - Painstaking manual detection process
  - Two bug reports = a pair of sequences of tokens
  - Label: Duplicates vs. non-duplicates
- Fraud
  - Sequence of actions performed by two accomplices
- Etc.

SMU SINGAPORE MANAGEMENT UNIVERSITY

Outline

- Motivation
- Definitions
- Mining Approach
  - Search Space Traversal
  - Tandem Projected Database
  - Pruning Strategies
  - Algorithms
- Experiments and Case Studies
- Conclusion and Future Work

Definitions

Labeled Sequence Pairs DB

- Labeled Sequence Pairs
  - Two series of events from an alphabet
  - With assigned label: +ve or –ve
- Example of a DB:

Idx	Sequence Pair	Label
1	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
2	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
3	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
4	$\langle a, a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
5	$\langle b, c, d, d\rangle - \langle e, f, g\rangle$	+ve
6	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	-ve
7	$\langle a, b, d, d\rangle - \langle e, d, c, d, e\rangle$	-ve
8	$\langle a, b, d, d\rangle - \langle c, d, d\rangle$	-ve
9	$\langle a, d, d\rangle - \langle e, c, d, e, d\rangle$	-ve

Dyadic Sequential Patterns

- Dyadic sequential pattern: Two sequences
- Support of pattern P=p1-p2
  - # of sequence pairs S=s1-s2 in DB, where:
    - p1 is a subsequence of s1 (or s2)
    - p2 is a subsequence of s2 (or s1)
  - sup_{+ve}/sup_{-ve}
- Discriminative score of P=p1-p2
  - Use information gain: IG(c|p) = H(c) – H(c|p)
  - A function of sup_{+ve} and sup_{-ve}

Dyadic Sequential Patterns

Idx	Sequence Pair	Label
1	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
2	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
3	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
4	$\langle a, a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
5	$\langle b, c, d, d\rangle - \langle e, f, g\rangle$	+ve
6	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	-ve
7	$\langle a, b, d, d\rangle - \langle e, d, c, d, e\rangle$	-ve
8	$\langle a, b, d, d\rangle - \langle c, d, d\rangle$	-ve
9	$\langle a, d, d\rangle - \langle e, c, d, e, d\rangle$	-ve

Num	Pattern P	sup(P)	disc(P)
1	$\langle a\rangle$ - $\langle d\rangle$	8	0.102
2	$\langle a, d\rangle$- $\langle d\rangle$	8	0.102
3	$\langle a, d, d\rangle$ - $\langle d\rangle$	8	0.102

SMU SINGAPORE MANAGEMENT UNIVERSITY

Closed Patterns

DEFINITION 3.12 **(Closed Pattern).** *A pattern p1 is closed if there does not exist another pattern p2 with the same support and discriminative score, where either one of the following conditions holds:*

$$(Cond\ 1)\ p1.Left \sqsubseteq p2.Left \wedge p1.Right \sqsubseteq p2.Right$$

$$(Cond\ 2)\ p1.Left \sqsubseteq p2.Right \wedge p1.Right \sqsubseteq p2.Left$$

Num	Pattern P	sup(P)	disc(P)
1	$\langle a \rangle$ - $\langle d \rangle$	8	0.102
2	$\langle a, d \rangle$- $\langle d \rangle$	8	0.102
3	$\langle a, d, d \rangle$ - $\langle d \rangle$	8	0.102

Subsumed By

$\langle a, d, d \rangle$ - $\langle d, d \rangle$	8	0.102

Problem Statement

- Given:
  - A dataset of labeled sequence pairs
  - Minimum support threshold
  - Minimum discriminative threshold
- Find a set of patterns which are:
  - Frequent
  - Discriminative
  - Closed

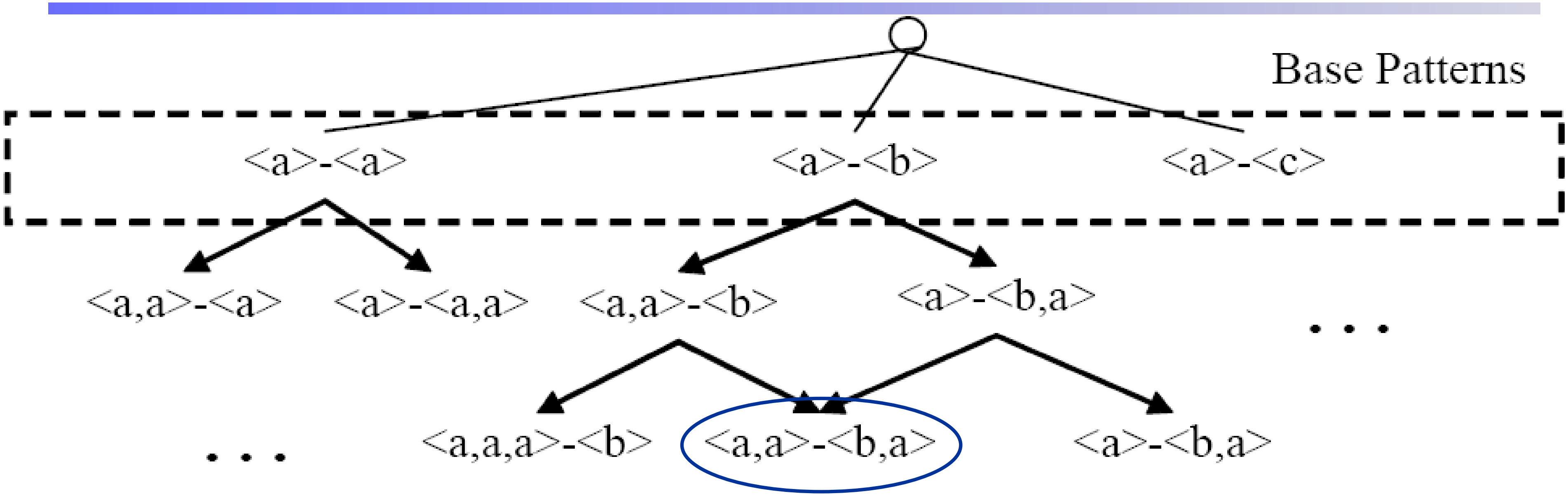
Mining Approach

Overall Strategy

- Traverse the search space of possible patterns
  - Ensure no important patterns are missed
  - Ensure no redundant visit
- Efficiently compute some statistics during traversal using a supporting data structure
  - Tandem projected database
- Prune search spaces containing:
  - Infrequent patterns
  - Non-discriminative patterns
  - Non-closed patterns

A. Search Space Traversal

Basic Search Space Traversal



- Start with base patterns (size=2)
- Grow base patterns
  - Append events to the left and right sequences
  - In depth first search fashion
- Problem: Redundant visits, e.g., <a,a>-<b,a>

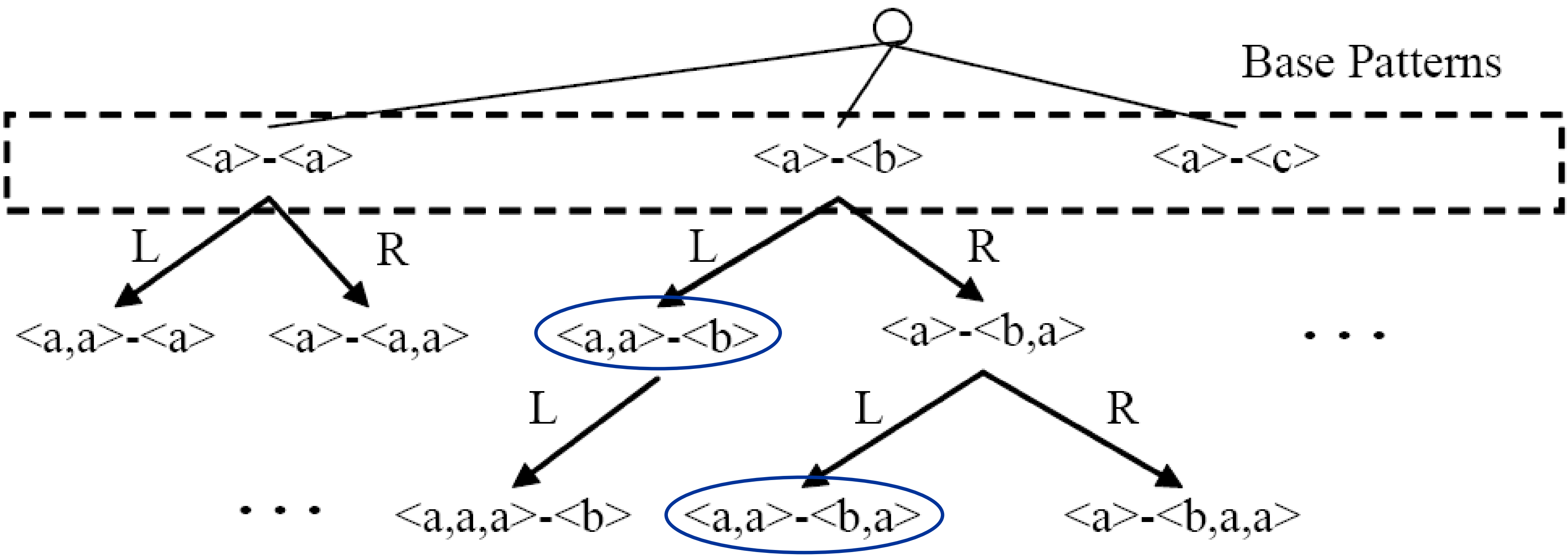
Handling redundant visits

- Definition: Left (right) extension of a pattern
  - Append an event to the left (right) sequence
- Label edges in the search lattice by L & R

- Prevent redundant visit
  - For every node visited via an L edge
  - Only L edges are traversed in subsequent growth operations

Handling redundant visits

- Why it works?
  - Every pattern could be formed,
  - by first performing right extensions,
  - followed by left extensions



Handling pattern isomorphism

- Some patterns are isomorphic
  - <a,b> - <c,d> is isomorphic to <c,d> - <a,b>
- Solution: introduce canonical patterns
  - Canonical: Left sequence <= right sequence
  - Based on a total ordering among events

PROPERTY 1 (**Canonical Pruning**). *A canonical left-extension pattern can only be grown from a canonical left- or right- extension pattern. A canonical right-extension pattern can only be grown from a canonical right-extension pattern.*

Overall Traversal Strategy

- Grow left-extension patterns leftwards
- Grow right-extension patterns in both directions
- Only output canonical patterns
- We do not need to grow non canonical patterns further

B. Tandem Projected DB

Tandem Projected Database

- Defined with respect to a dyadic pattern
- Suffixes of the pairs of sequences in DB whose prefixes match the pattern

- Represented as a set of 4 numbers [(a,b),(c,d)]
  - a & b represent the 2 suffixes when: L -> L & R -> R
  - c & d represent the 2 suffixes when: L -> R & R -> L

- Implemented as a set of 2 simple PDB entries
  - One representing (a,b) and another representing (c,d)
  - Tied one after another (in tandem)

Tandem Projected Database

Idx	Sequence Pair	Label
1	$\langle a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	+ve
2	$\langle a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	+ve
3	$\langle a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	+ve
4	$\langle a,a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	+ve
5	$\langle b,c,d,d\rangle - \langle e,f,g\rangle$	+ve
6	$\langle a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	-ve
7	$\langle a,b,d,d\rangle - \langle e,d,c,d,e\rangle$	-ve
8	$\langle a,b,d,d\rangle - \langle c,d,d\rangle$	-ve
9	$\langle a,d,d\rangle - \langle e,c,d,e,d\rangle$	-ve

- Projected database of <a,d>-<c,d> in sequence 1 above, i.e., <a,b,d,d>-<e,c,d,d,e> is:
  - [(<d>,<d,e>),(ε, ε)]

SMU SINGAPORE MANAGEMENT UNIVERSITY

C. Pruning Properties

Pruning Properties

PROPERTY 2 **(Anti-Monotonicity of Support).** *The support of a pattern P is always greater than or equal to the support of its descendants.*

PROPERTY 3 **(Upper Bound of Discrimin. Score).** *For pattern P and database DB, disc(P,DB) is bounded by:*

$$disc_{ub}(P) = \max(IG(sup_{+ve}(P), 0), IG(0, sup_{-ve}(P)))$$

We denote the upper bound on the discriminative score of a pattern P as $disc_{ub}(P)$.

PROPERTY 4 **(Anti-Monotonicity of Disc. Bound).** *For pattern P and its descendant P′, $disc_{ub}(P) \geq disc_{ub}(P')$.*

In-Between Event Sets

- Consider a pattern P=p1-p2 and a sequence pair S containing it.
- There are |p1|+|p2| in-between event sets.
- Informally, they are:
  - Events in s which appear between the occurrences of two consecutive events in P
  - Or before the occurrences of the first events of P
- Two variants:
  - (Regular) In-Between Event Sets
  - Strict In-Between Event Sets

In-Between Event Sets

Idx	Sequence Pair	Label
1	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
2	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
3	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
4	$\langle a, a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	+ve
5	$\langle b, c, d, d\rangle - \langle e, f, g\rangle$	+ve
6	$\langle a, b, d, d\rangle - \langle e, c, d, d, e\rangle$	-ve
7	$\langle a, b, d, d\rangle - \langle e, d, c, d, e\rangle$	-ve
8	$\langle a, b, d, d\rangle - \langle c, d, d\rangle$	-ve
9	$\langle a, d, d\rangle - \langle e, c, d, e, d\rangle$	-ve

- Consider pattern <a>-<e,c,e> and the 1st sequence
  - Event d could be inserted in-between c & e
  - d is in the in-between event set R_3 for S1

Closed Pattern Properties

DEFINITION 6.5 (**Forward Extension**). *A forward extension event of a pattern P is an event that could be appended to P (i.e., any sequence of P) resulting in another pattern with the same support.*

DEFINITION 6.6 (**Backward Extension**). *A backward extension of a pattern P is an event that could be inserted to P (i.e., any sequence of P) resulting in another pattern with the same support.*

Closed Pattern Properties

PROPERTY 6 (**Backward Extension Set**). *The backward extension set of a pattern P are events appearing in one of the in-between event sets of P in all sequence pairs supporting P in the database. Mathematically, this is the set:*

$$\{e | \exists x \in \{L_1, \ldots, L_{|P.Left|}, R_1, \ldots, R_{|P.Right|}\}. \forall_{(S \in DB) \wedge (P \sqsubseteq S)}.e \in x(P,S)\}$$

PROPERTY 8 (**Closure Check**). *If a pattern has no forward extension and no backward extension, then it is closed.*

Closed Pattern Properties

Idx	Sequence Pair	Label
1	$\langle a, b, d, d \rangle - \langle e, c, d, d, e \rangle$	+ve
2	$\langle a, b, d, d \rangle - \langle e, c, d, d, e \rangle$	+ve
3	$\langle a, b, d, d \rangle - \langle e, c, d, d, e \rangle$	+ve
4	$\langle a, a, b, d, d \rangle - \langle e, c, d, d, e \rangle$	+ve
5	$\langle b, c, d, d \rangle - \langle e, f, g \rangle$	+ve
6	$\langle a, b, d, d \rangle - \langle e, c, d, d, e \rangle$	-ve
7	$\langle a, b, d, d \rangle - \langle e, d, c, d, e \rangle$	-ve
8	$\langle a, b, d, d \rangle - \langle c, d, d \rangle$	-ve
9	$\langle a, d, d \rangle - \langle e, c, d, e, d \rangle$	-ve

- Consider pattern P = <a,b,d,d>-<e,c,d,d,e>
  - It has no forward or backward extension
  - It is closed

Closed Pattern Properties

PROPERTY 9 (**Non-Closedness Pruning**). *If there is an event in one of P strict in-between event sets for all sequences containing P in DB, then P and all descendants of P are not closed.*

SMU SINGAPORE MANAGEMENT UNIVERSITY

Closed Pattern Properties

Idx	Sequence Pair	Label
1	$\langle a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	+ve
2	$\langle a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	+ve
3	$\langle a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	+ve
4	$\langle a,a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	+ve
5	$\langle b,c,d,d\rangle - \langle e,f,g\rangle$	+ve
6	$\langle a,b,d,d\rangle - \langle e,c,d,d,e\rangle$	-ve
7	$\langle a,b,d,d\rangle - \langle e,d,c,d,e\rangle$	-ve
8	$\langle a,b,d,d\rangle - \langle c,d,d\rangle$	-ve
9	$\langle a,d,d\rangle - \langle e,c,d,e,d\rangle$	-ve

- Consider pattern P = <a>-<e,c,e>
  - Event d could be inserted in-between c & e
    - For all sequence pairs supporting P
  - P and all its descendants are not closed

D. Algorithms

Algorithm 1: Baseline.

1. Consider the left & right sequences of the pairs separately. Create a standard sequence DB.
2. Mine standard frequent sequential patterns.
3. Pair up all mined frequent sequential patterns.
4. Compute the support and discriminative score of each of the resultant pairs.
5. Output those that are frequent and discriminative.

Algorithm 2: Mine All Frequent Disc. Patterns

Procedure MineAllFrequent

Inputs:

DB : Database of sequence pairs

min_sup : Minimum support threshold

min_disc: Minimum discriminative threshold

Output:

All patterns that are frequent and discriminative

Methods:

1: Let Base = Canonical & frequent base patterns with $disc_{ub} \geq min\_disc$

2: Compute tandem projected db for patterns $\in$ Base

3: For each p in Base

4: Grow(p, "LR", min_sup, min_disc)

Procedure Grow (pattern p, L/LR ext. Dir, thresh.)

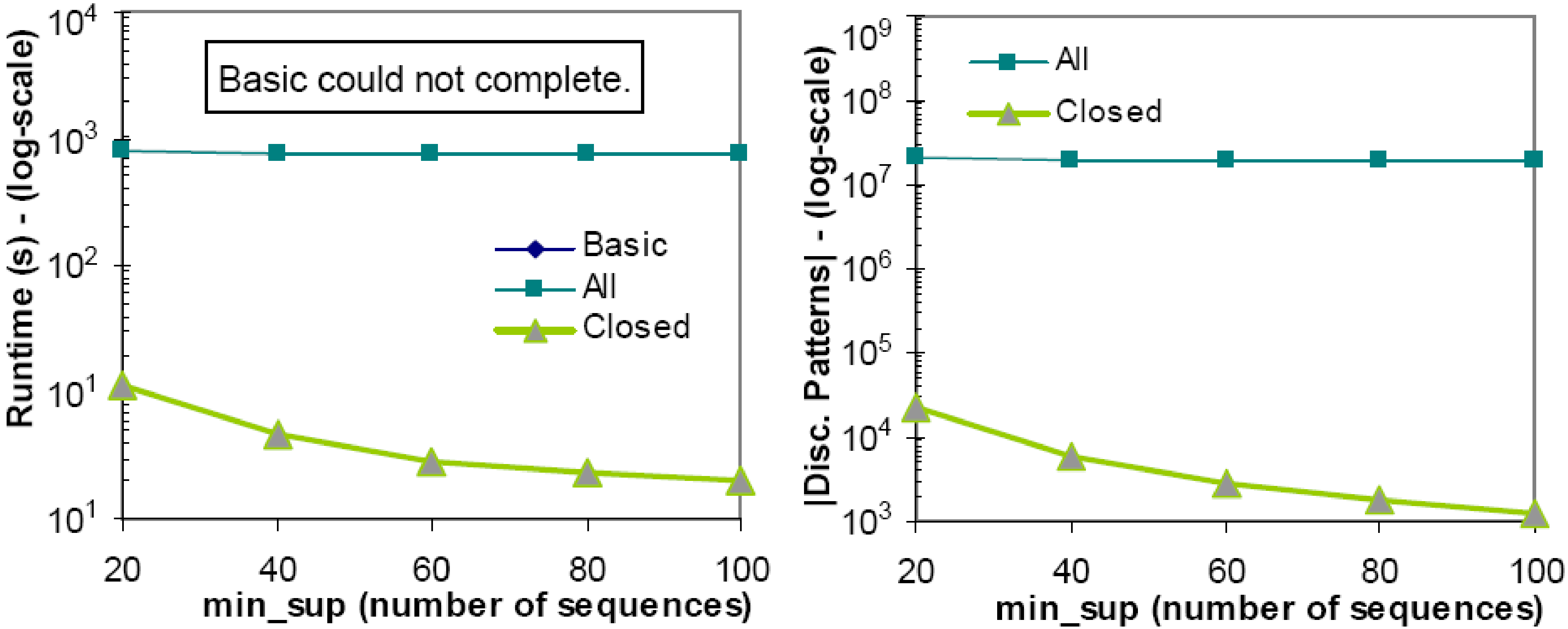
If (disc(p) $\geq min\_disc$)
    Output p
Let PDB = projected database of p
// Grow Left
Let LFE_L = {ev|exists $\geq min\_sup$ entries [(a,b),(c,d)] in PDB with ev $\in$ a or $ev \in$ c}
For each event e_L in LFE_L
    Let p' = ($p1$++ e_L)-$p2$
    If p' is canonical
        Compute projected database of p' from PDB
        If (sup(p') $\geq min\_sup \wedge$ disc$_{ub}$(p') $\geq min\_disc$)
            Grow(p', "L", min_sup, min_disc)
// Grow Right
If (Dir="LR")
...
            Grow(p', "LR", min_sup, min_disc)

Algorithm 3: Mine Closed Patterns

```
6:   If (disc(p) ≥ min_disc ∧ p satisfies Property 8)
7:        Output p
...
// Grow Left
...
14:            If (sup(p') ≥ min_sup ∧ disc_ub(p') ≥ min_disc)
c1:                 If (p' is not prunable by Property 9)
15:                      Grow(p', "L", min_sup, min_disc)
// Grow Right
...
22:            If (sup(p') ≥ min_sup ∧ disc_ub(p') ≥ min_disc)
c2:                 If (p' is not prunable by Property 9)
23:                      Grow(p', "LR", min_sup, min_disc)
```

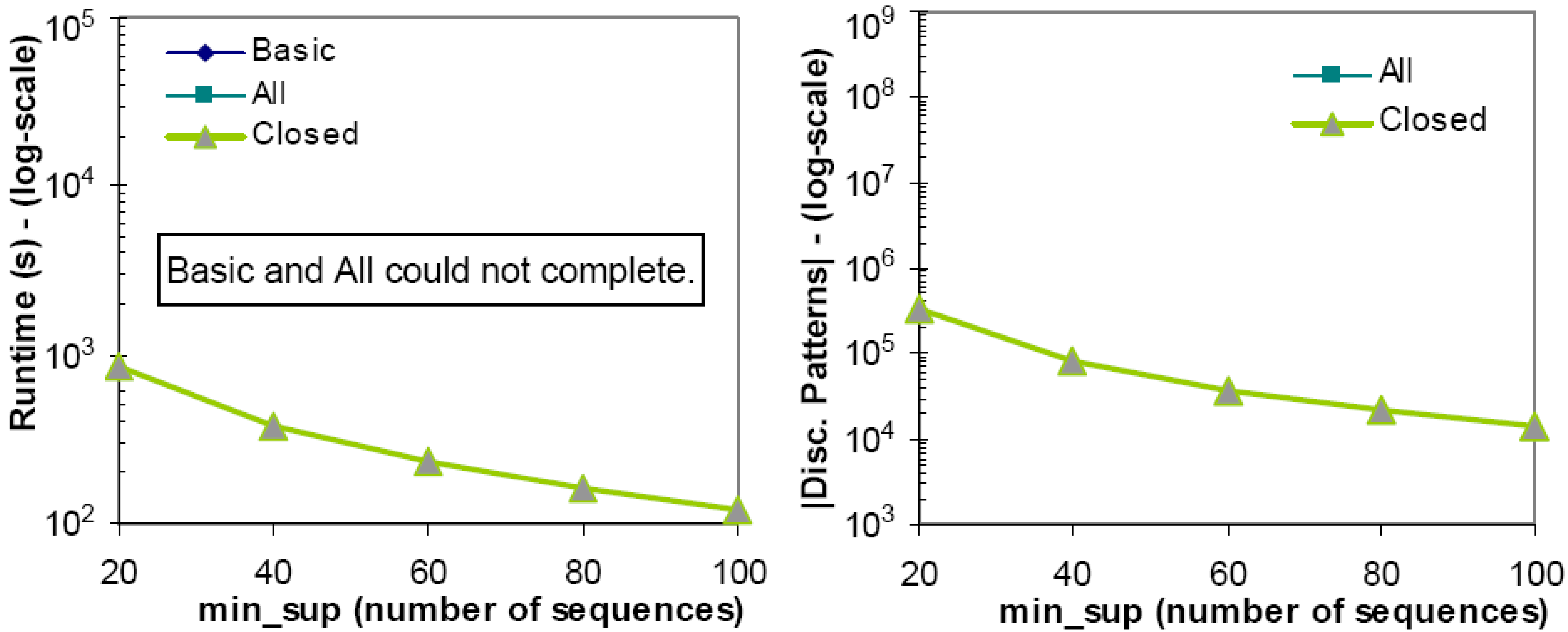
Experiments and Case Studies

Experiments



- [Synthetic Data] D = 10k, PNum = 10, PSize = 30

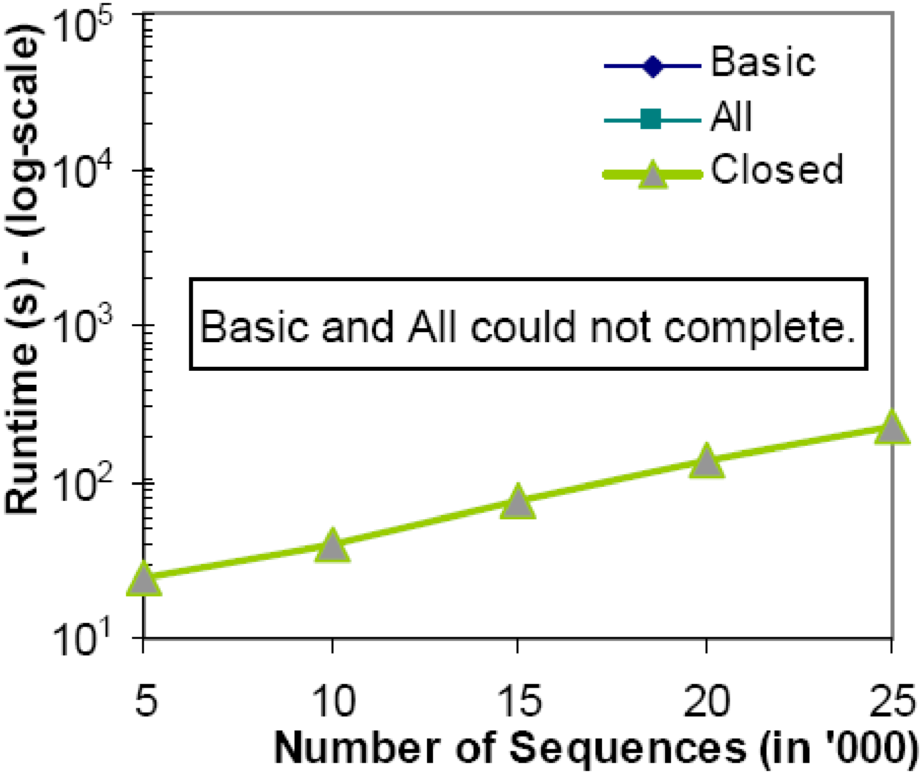
Experiments

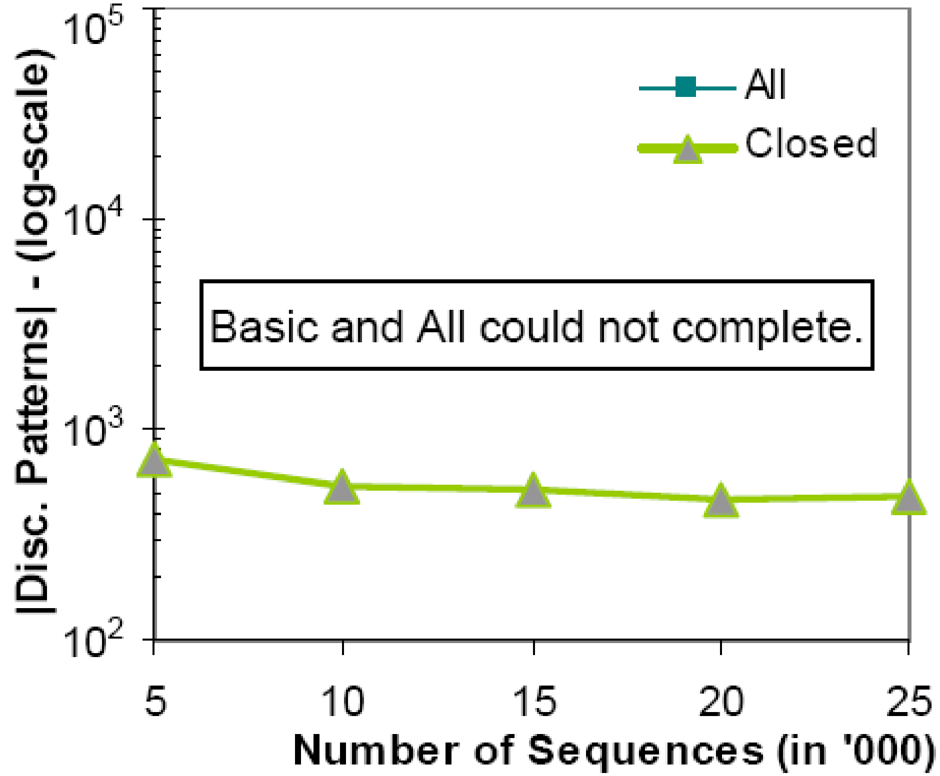


- [Synthetic Data] D = 25k, PNum = 30, PSize = 30

SMU SINGAPORE MANAGEMENT UNIVERSITY

Experiments





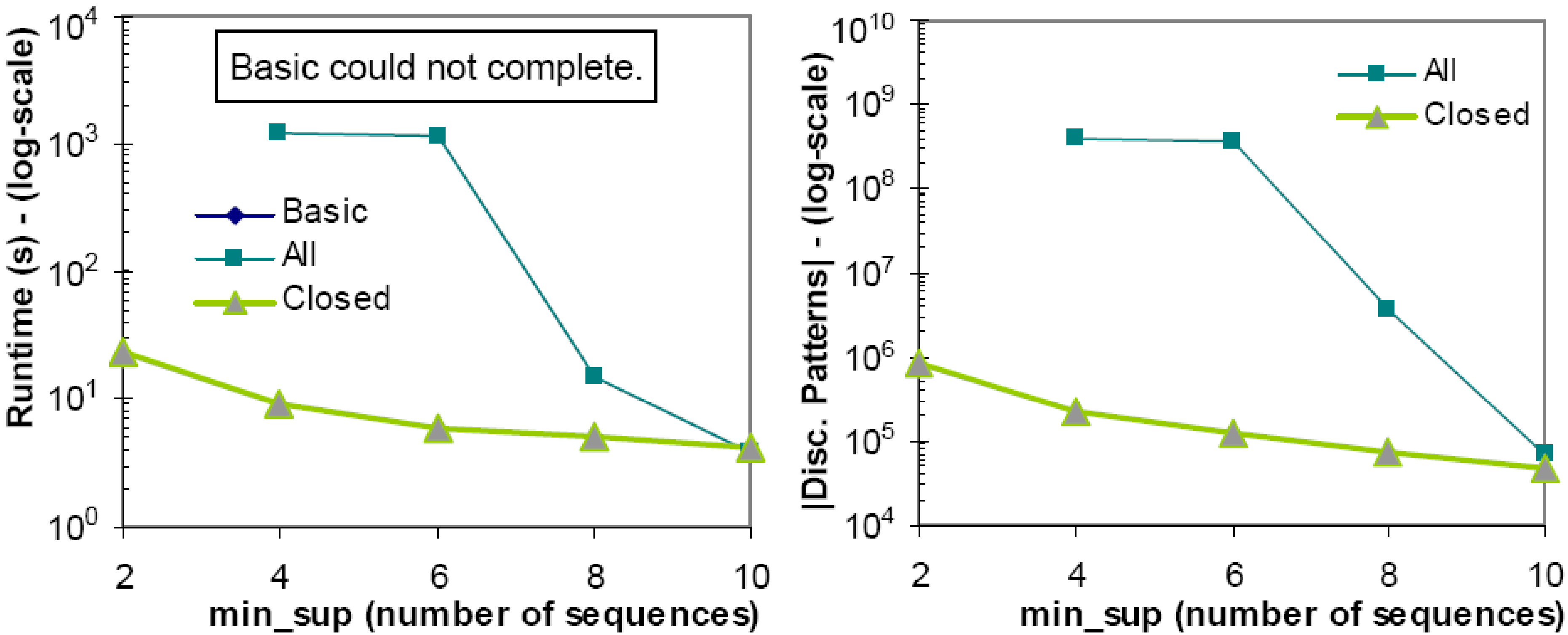
- [Synthetic Data] min_sup = 60, PNum = 30, PSize = 30

SMU SINGAPORE MANAGEMENT UNIVERSITY

Real dataset

- Raw bug reports
  - 12,732 bug reports from OpenOffice
  - 44,652 bug reports from Eclipse
  - 47,704 bug reports from Firefox
- Find historical bug report duplicate pairs
  - 5,949 duplicate pairs
- Create non duplicate bug report pairs
  - 5,949 non duplicate pairs
- Total
  - 11,898 pairs with 8,601 different events
  - Average size: 13.75 events; Largest: 62 events

Experiments



- [Real Dataset: Bug Reports Data]

SMU SINGAPORE MANAGEMENT UNIVERSITY

Case Study

- Task: Predict if a pair of bug reports are duplicates of each other or not.
- Settings:
  - Use LibSVM as a classification engine
  - Single tokens: Set of tokens appearing in a pair.
  - Dyadic patterns: Mined patterns (min_sup=2, min_disc=0.0001)

Configuration	Accuracy	AUC
Single Tokens	60.38%	0.65
Dyadic Patterns	82.86%	0.90
Both	81.23%	0.89

Table 4: Accuracy: Duplicate Bug Report Detection

SINGAPORE MANAGEMENT UNIVERSITY

Conclusion

- Propose a new problems of mining dyadic sequential patterns
  - Frequent, closed, discriminative
- Employ new:
  - Search space traversal strategy
  - Data structure
  - Pruning properties
- Achieve more than 2 orders of magnitude faster
- Increase accuracy from 60% to 82% and AUC from 0.65 to 0.90 on a real bug report dataset.

Future Work

- Experiment on more datasets
  - Further demonstrate the power of dyadic patterns,
  - as good features for classification purpose
- Improve the efficiency further
- Improve the expressiveness of the patterns
  - Triadic sequential patterns
  - Multi-adic sequential patterns
  - Pairs of sequences of sets

Acknowledgement

- We would like to thank the anonymous reviewers for their valuable comments and advice.

Thank You

Questions, Comments, Advice ?