Mining Adaptively Frequent Closed Unlabeled Rooted Trees in Data Streams

Albert Bifet and Ricard Gavaldà

Universitat Politècnica de Catalunya

14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)
2008 Las Vegas, USA

Data Streams

- Sequence is potentially infinite
- High amount of data: sublinear space
- High speed of arrival: sublinear time per example

Tree Mining

- Mining frequent trees is becoming an important task
- Applications:
  - chemical informatics
  - computer vision
  - text retrieval
  - bioinformatics
  - Web analysis.
- Many link-based structures may be studied formally by means of unordered trees

Introduction: Trees

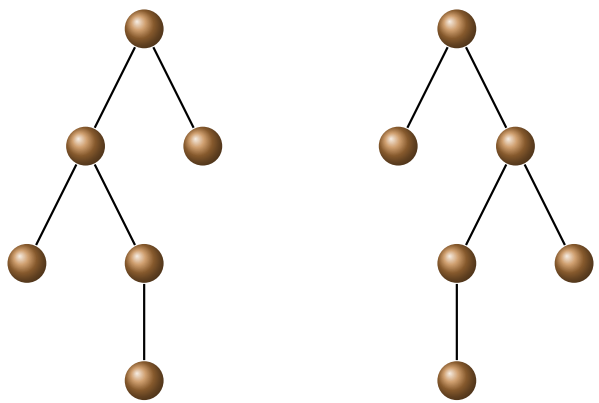
Our trees are:

- Rooted
- Unlabeled
- Ordered and Unordered

Our subtrees are:

- Induced

Two different ordered trees
but the same unordered tree



What Is Tree Pattern Mining?

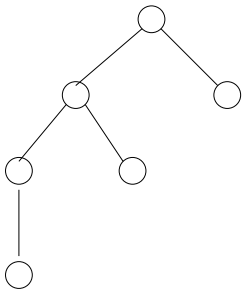
Given a dataset of trees, find the complete set of frequent subtrees

- Frequent Tree Pattern (FS):
  - Include all the trees whose support is no less than min_sup
- Closed Frequent Tree Pattern (CS):
  - Include no tree which has a super-tree with the same support
- $CS \subseteq FS$
- **Closed Frequent Tree Mining** provides a **compact** representation of frequent trees without loss of information

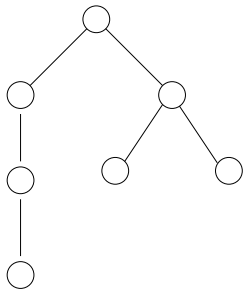
Introduction

Unordered Subtree Mining

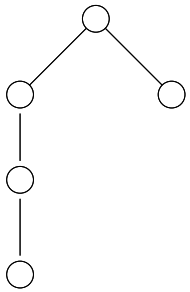
A:



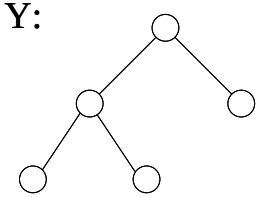
B:



X:



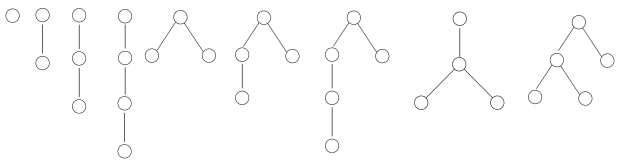
Y:



$$D = \{A, B\}, min\_sup = 2$$

Closed Subtrees : 2
Frequent Subtrees: 9

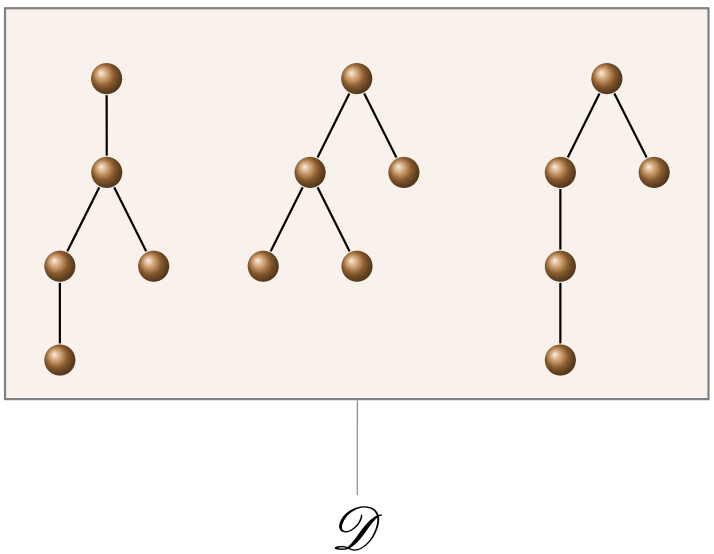
- Closed Subtrees: X, Y
- Frequent Subtrees:



Introduction

Problem

Given a data stream $\mathscr{D}$ of rooted, unlabelled and unordered trees, find frequent closed trees.



$\mathscr{D}$

We provide three algorithms, of increasing power

- Incremental
- Sliding Window
- Adaptive

Relaxed Support

Guojie Song, Dongqing Yang, Bin Cui, Baihua Zheng, Yunfeng Liu and Kunqing Xie.
CLAIM: An Efficient Method for Relaxed Frequent Closed Itemsets Mining over Stream Data

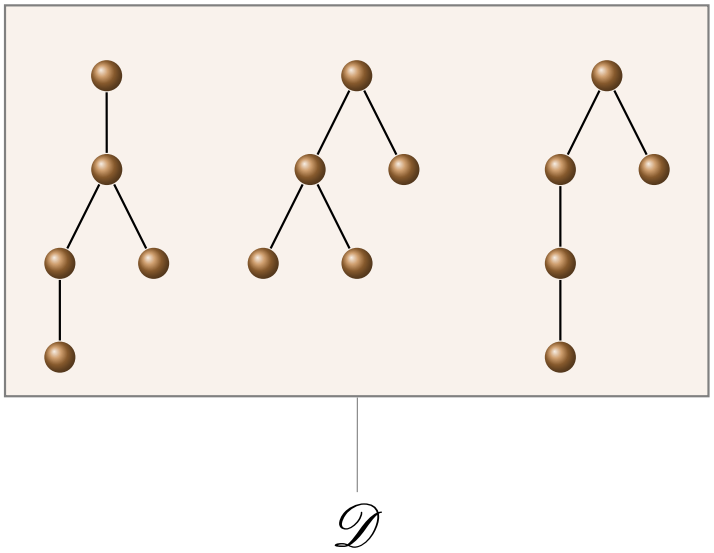
- **Linear Relaxed Interval**:The support space of all subpatterns can be divided into $n = \lceil 1/\varepsilon_r \rceil$ intervals, where ε_r is a user-specified relaxed factor, and each interval can be denoted by $\mathscr{I}_i = [l_i, u_i)$, where $l_i = (n-i) * \varepsilon_r \geq 0$, $u_i = (n-i+1) * \varepsilon_r \leq 1$ and $i \leq n$.
- **Linear Relaxed closed subpattern** t: if and only if there exists no proper superpattern t' of t such that their suports belong to the same interval $\mathscr{I}_i$.

Relaxed Support

As the number of closed frequent patterns is not linear with respect support, we introduce a new relaxed support:

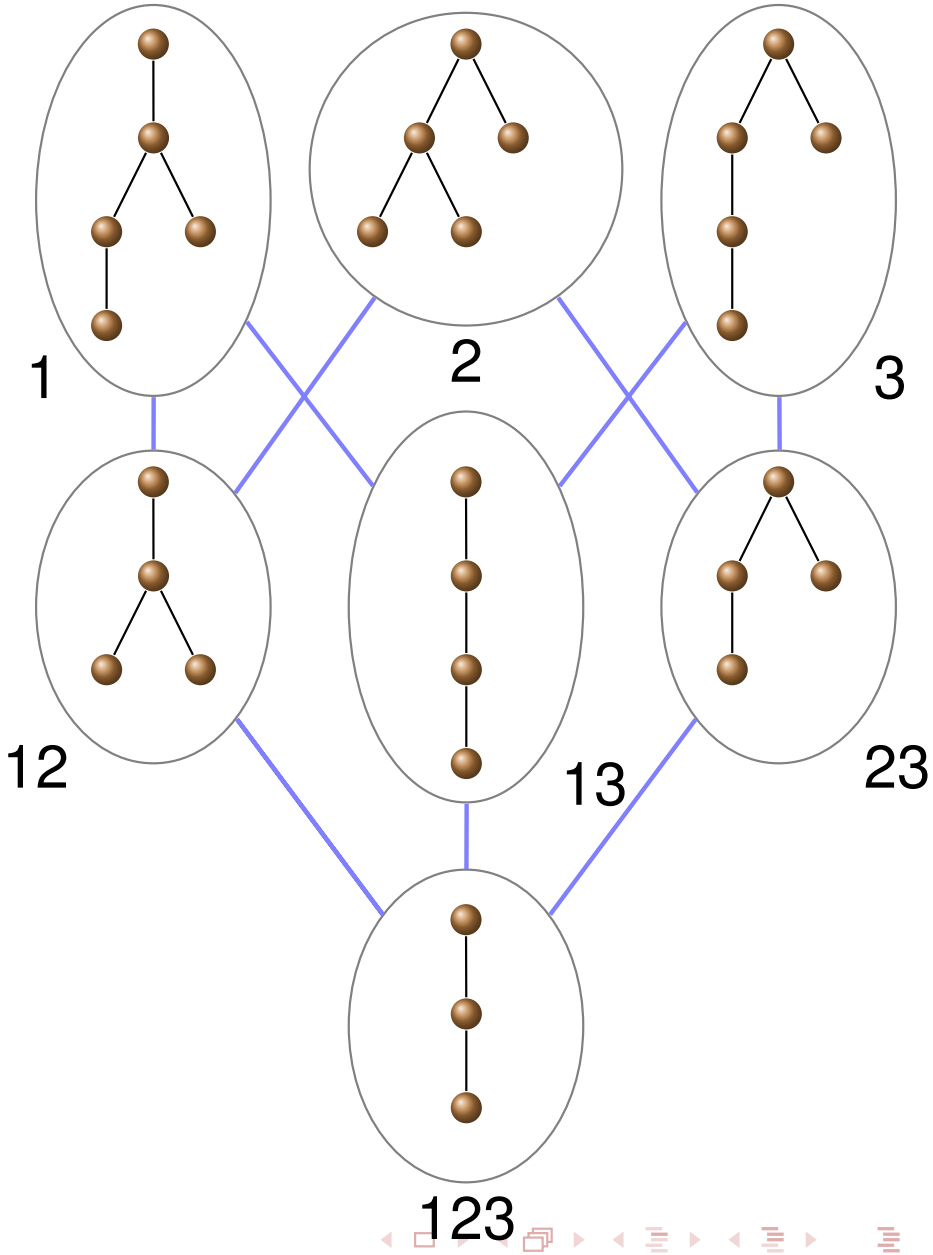
- **Logarithmic Relaxed Interval**:The support space of all subpatterns can be divided into $n = \lceil 1/\varepsilon_r \rceil$ intervals, where ε_r is a user-specified relaxed factor, and each interval can be denoted by $\mathscr{I}_i = [l_i, u_i)$, where $l_i = \lceil c^i \rceil$, $u_i = \lceil c^{i+1} - 1 \rceil$ and $i \leq n$.
- **Logarithmic Relaxed closed subpattern** t: if and only if there exists no proper superpattern t' of t such that their suports belong to the same interval $\mathscr{I}_i$.

Galois Lattice of closed set of trees



We need

- a Galois connection pair
- a closure operator



Algorithms

Algorithms

- Incremental: IncTreeNat
- Sliding Window: WinTreeNat
- Adaptive: AdaTreeNat Uses `ADWIN` to monitor change

`ADWIN`

An adaptive sliding window whose size is recomputed online according to the rate of change observed.

`ADWIN` has rigorous guarantees (theorems)

- On ratio of false positives and negatives
- On the relation of the size of the current window and change rates

Experimental Validation: TN1

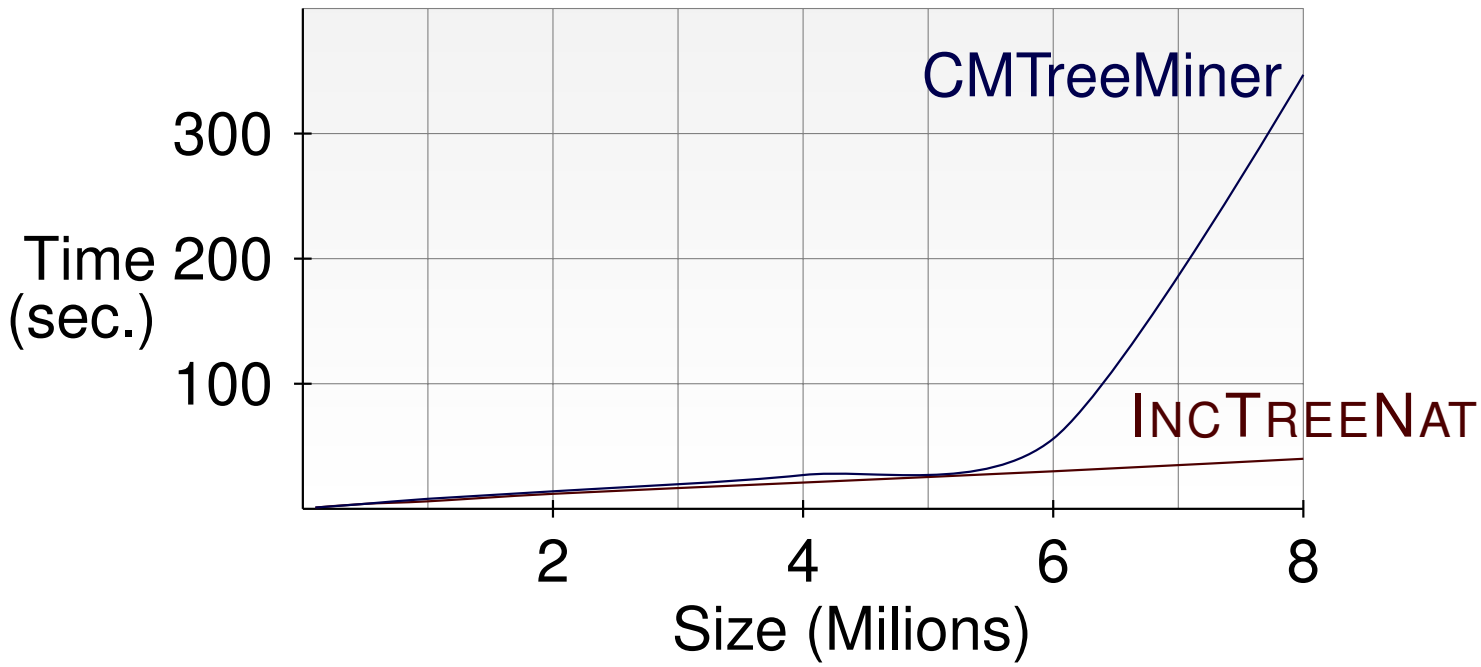


Figure: Time on experiments on ordered trees on TN1 dataset

Experimental Validation

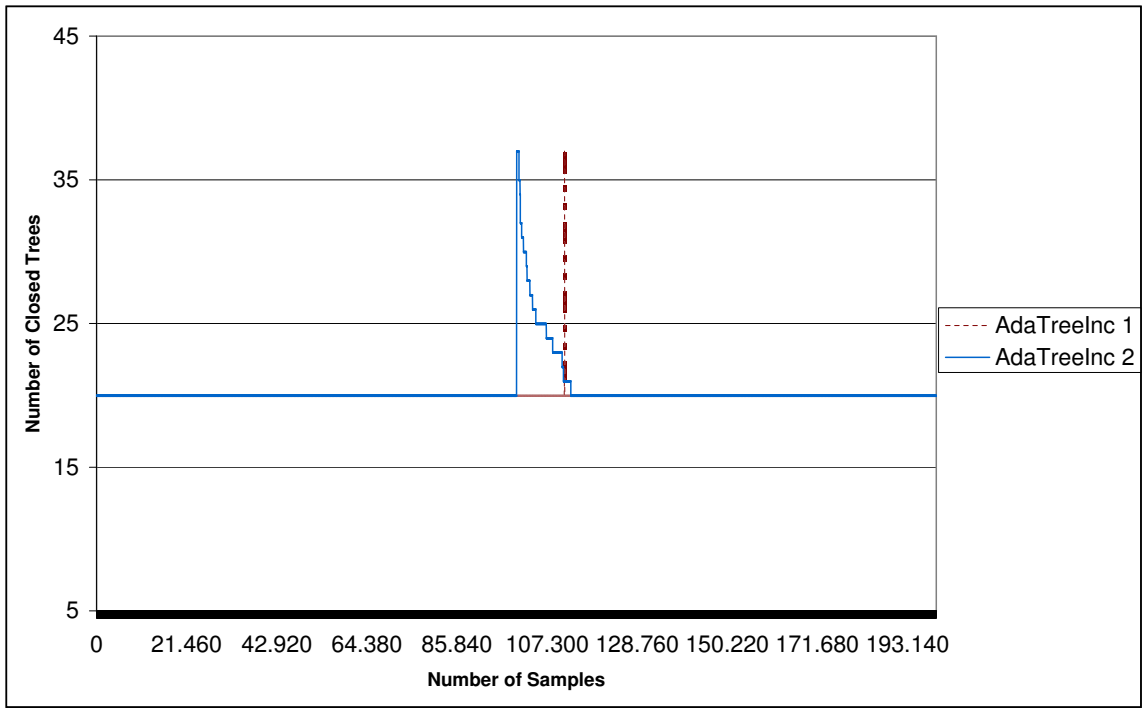


Figure: Number of closed trees maintaining the same number of closed datasets on input data

Summary

Conclusions

- New logarithmic relaxed closed support
- Using Galois Latice Theory, we present methods for mining closed trees
  - Incremental: IncTreeNat
  - Sliding Window: WinTreeNat
  - Adaptive: AdaTreeNat using ADWIN to monitor change

Future Work

Labeled Trees and XML data.