Mining Frequent Closed Graphs on Evolving Data Streams

A. Bifet, G. Holmes, B. Pfahringer and R. Gavaldà

University of Waikato
Hamilton, New Zealand

Laboratory for Relational Algorithmics, Complexity and Learning, LARCA
UPC-Barcelona Tech

1st Graph-TA Workshop, Feb. 19th, 2013
Originally presented at KDD’11
Mining Evolving Graph Data Streams

Problem

Given a stream $\mathcal{G}$ of graphs, maintain the set of frequent closed subgraphs
## Graph Dataset

<table>
<thead>
<tr>
<th>Transaction Id</th>
<th>Graph</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>O C - C - S - N O</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>O C - C - S - N C</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>O C - S - N C</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>N C - C - S - N</td>
<td>1</td>
</tr>
</tbody>
</table>
Frequent Closed Pattern Mining

- Universe $U$ of patterns
- Subpattern partial order, denoted $P \leq P'$
- Support of a pattern $P$ in a multiset $\mathcal{D} = \ldots$
  - fraction of $\mathcal{D}$ elements that are have $P$ as subpattern
- Pattern $P$ is closed in $\mathcal{D}$ if every superpattern of $P$ has smaller support

The frequent closed pattern mining problem

Given $\mathcal{D}$, find the set of closed patterns with support $\geq \varepsilon$
The Data Stream Computation Model

Five constraints:

1. Input is sequence of items; $t$-th item available at time $t$
2. Answers must be anytime, may be approximate
3. Low processing time per item
4. Sublinear memory; keep only summaries or sketches
5. Data distribution evolves over time; forget, react, adapt
Previous work

- *CloseGraph* [Yan-Han 03]
  - depth-first search, based on gSpan ICDM’02
- *MoSS* [Borgelt-Berthold 05]
  - breadth-first search, based on MoFa ICDM’02

**Non-streaming**: Non-incremental, multipass, linear memory
**Graph Coresets**

**Coreset of a set $P$ with respect to some problem**

Small subset that approximates the original set $P$

- Solving the problem for the coreset provides an approximate solution for the problem on $P$

**$\delta$-tolerance Closed Graph**

A graph $g$ is $\delta$-tolerance closed if none of its proper frequent supergraphs has a weighted support \( \geq (1 - \delta) \cdot \text{support}(g) \)

- Maximal graph: 1-tolerance closed graph
- Closed graph: 0-tolerance closed graph
Graph Coresets

**Relative support of a closed graph**
Support of a graph minus the relative support of its closed supergraphs

- The sum of the closed supergraphs’ relative supports of a graph and its relative support is equal to its own support

**$(s, \delta)$-coreset for computing closed graphs**
Weighted multiset of frequent $\delta$-tolerance closed graphs with minimum support $s$ using their relative support as a weight
Dealing with evolution over time

- Keep a window on recent stream elements
  - Actually, just its lattice of closed elements!
- Keep track of number of closed trees in lattice, \( N \)
- Use some change detector on \( N \)
- When change is detected:
  - Drop stale part of the window
  - Update lattice to reflect this deletion, using deletion rule

Alternatively, sliding window of some fixed size
**WinGraphMiner**

**WinGraphMiner**$(D, W, \text{min}\_sup)$

Input: A graph dataset $D$, a size window $W$ and $\text{min}\_sup$.
Output: The frequent graph set $G$.

1. $G \leftarrow \emptyset$
2. for every batch $b_t$ of graphs in $D$
3.     do $C \leftarrow \text{CORESET}(b_t, \text{min}\_sup)$
4.     Store $C$ in sliding window
5.     if sliding window is full
6.         then $\overline{R} \leftarrow$ Oldest $C$ stored in sliding window, negate all support values
7.     else $\overline{R} \leftarrow \emptyset$
8. $G \leftarrow \text{CORESET}(G \cup C \cup \overline{R}, \text{min}\_sup)$
9. return $G$
Experimental Evaluation

ChemDB dataset
- Public dataset
- 4 million molecules
- Institute for Genomics and Bioinformatics at the University of California, Irvine

Open NCI Database
- Public domain
- 250,000 structures
- National Cancer Institute
Open NCI dataset

Time NCI Dataset

Instances

Seconds

- IncGraphMiner
- IncGraphMiner-C
- MoSS
- closeGraph
Open NCI dataset

Memory NCI Dataset

Instances

Megabytes

IncGraphMiner
IncGraphMiner-C
MoSS
closeGraph
ChemDB dataset

Memory ChemDB Dataset

Instances

Megabytes

IncGraphMiner
IncGraphMiner-C
MoSS
closeGraph
ChemDB dataset
Summary

We provide three algorithms of increasing power:

- Incremental
- Sliding Window
- Adaptive

To our knowledge, first algorithms for mining frequent (closed) subgraphs from evolving data streams