Top-*k* Event Discovery for Graph Streams

Shayan Monadjemi[1], Mohammad Hossein Namaki[2], Yinghui Wu[2], Lawrence Holder[2]

[1] The University of Texas at Dallas, Richardson, TX 75080

[2] Washington State University, Pullman, WA 99164

Smart Environments REU Program

Introduction

- Given an evolving network (e.g., social network, cyber network, brain connectome), how to extract and monitor top-*k* significant complex events? This project studies top-*k* event discovery problem in a graph stream.
- We define metrics to measure the interestingness of events in graph streams and developed a graph stream mining algorithm to discover top-*k* events.
- Using real world graph streams such as the Panama Dataset[1], we experimentally verify the efficiency of our algorithms.

[1] International Consortium of Investigative Journalists: https://offshoreleaks.icij.org/

Motivation

- Discovering and monitoring the top-*k* most frequent events in data graphs will helps us learn about the evolution of a graph as well as the most trending patterns.
- For example, Denial of Service (DDos) is a technique attackers use to hack computers on a network by continuously sending requests to computers on a network. Such masked attacks in a graph representing a computer network (*G*) can be described by the two attack patterns P_1 and P_2 in figure 1.
- In a Smart Environment setting, the graph could contain information about the inhabitant's activities. Performing the top-*k* event discovery on such data graph will help the system with activity recognition.

Problem Definition

Given an initial data graph, G0, and a continuous stream of edge additions/deletions, ΔE, we want to find the top-k most frequent patterns in our evolving data graph. A pattern captures the neighborhood similarity between the pattern node and its match. (Figure 1)

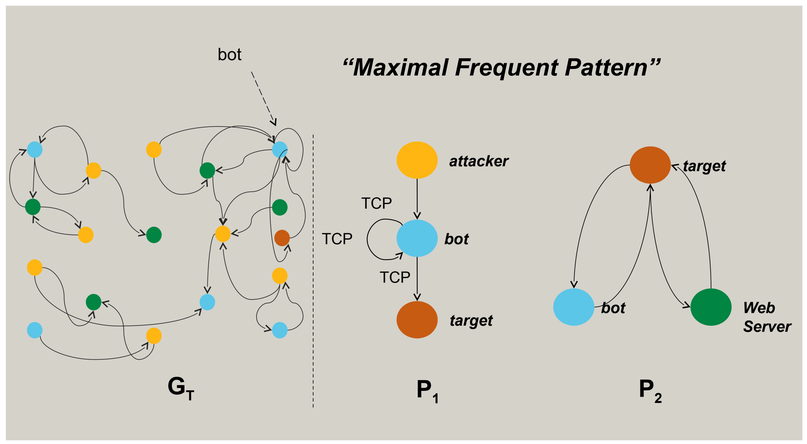

Figure 1: An example of data graph and event patterns

We introduce a weighted support as follows. (1) We define the weight of occurrence $\mathrm{occ}(P, t_i)$ as:

$$\omega_i^P = \alpha^{T-i} \frac{|occ(P(\bar{u}), t_i)|}{|V_{t_i}|}$$

Where $\alpha \in (0,1]$ refers to a decay factor posed on the times-temp. (2) The support will be:

$$\mathrm{supp}(P, G_T) = \sum \omega_i^P \quad , \qquad i \in [1, \mathrm{T}].$$

The following is the definition of our algorithm:

- **Input:** (1) An initial data graph, *G0*; (2) A stream of edge addition/deletions, *ΔE*; (3) Integer *k*, the number of top patterns to get; (4) A support threshold, ϑ, the minimum support of a pattern to be considered as a candidate.
- **Output:** A top-*k* event pattern set $\Sigma = \{P_1, \dots, P_k\}$, such that (1) each event pattern $P_i \in \Sigma$ is a maximal frequent pattern with respect to ϑ, and (2) the total support $\sum_{P_i \in \Sigma} \mathrm{Supp}(P_i)$ is maximized.

Top-*k* Discovery Framework

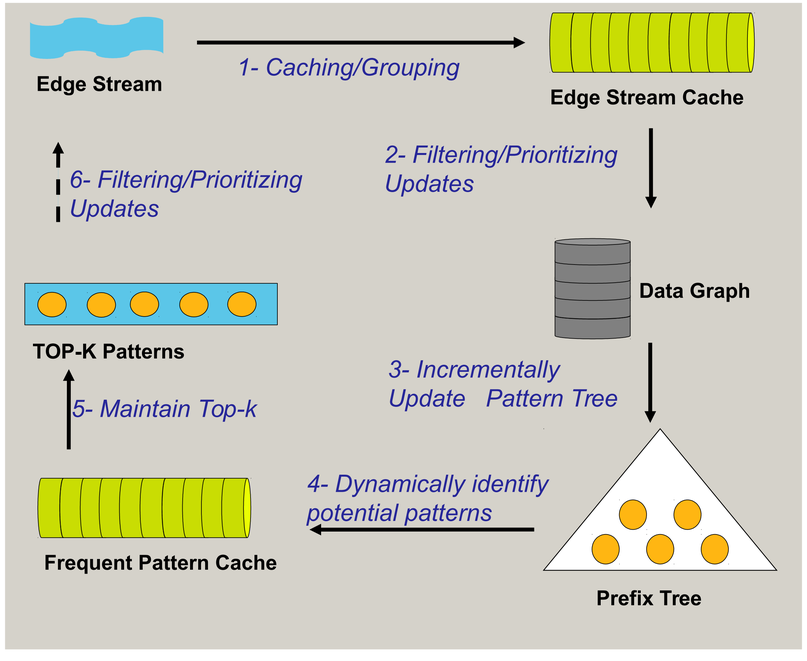

Figure 2: The cycle of Top-*k* event discovery algorithm

Pattern Tree Maintenance

Steps (1), (2), and (3) in figure 2 represent the generation and maintenance of the pattern tree.

- **Pipelining:** Prioritize the edge streams that affect the top-*k* result, and defer the process of unimportant updates.
- **Grouping:** Group the edge streams based source/target, and cancel out transactions that would be add up to be neutral. Incrementally process groups of updates.
- **Early Termination:** The generation of the prefix tree stops when we have found top-k event patterns that we consider "frequent and interesting," using a threshold.

Case Study

We examined our algorithm on the Panama Offshore Dataset with two different settings. The points of interest for the first experiment was the entities that are in the U.S. and have invested in foreign tax havens. The results that we got indicated that British Virgin Islands, Panama, and Bahamas were the top three tax havens for such entities. For the second experiment, we added Status_Inactivated as an additional focus node (i.e. point of interest). The results indicate that Panama, British Virgin Islands, Nevada are the top 3 jurisdictions with such inactivated entities.

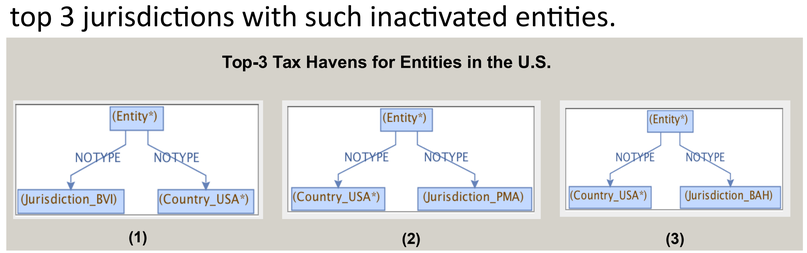

Figure 3: Event patterns (1),(2), and (3) have 1564, 794, and 402 matches respectively.

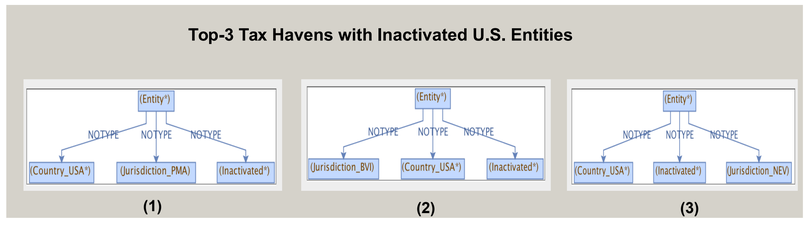

Figure 4: Event patterns (1),(2), and (3) have 48, 19, and 11 matches respectively.

Performance

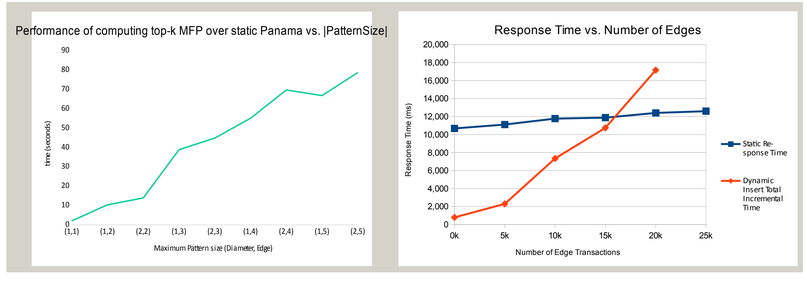

Figure 5: The performance of the algorithm from different perspectives. (1) Left: The runtime w.r.t the diameter of the requested patterns (hops and edges). (2) Comparing response time of incremental algorithm with batch algorithm. Graph databases from the Panama dataset each started with 2 million initial relationships. The focus node had 3215 candidates.

Future Work

In the future versions of this algorithm, we will apply optimization techniques to improve stream mining algorithms. Furthermore, we will add the functionality of discovering and monitoring top-*k* event patterns over a sliding window rather than a single window.

Acknowledgements

This material is based upon work supported by the National Science Foundation Research Experiences for Undergraduates Program under Grant No. 1460917.