HashNWalk: Hash and Random Walk Based Anomaly Detection in Hyperedge Streams

Geon Lee, Minyoung Choe, and Kijung Shin
KAIST
{geonlee0325, minyoung.choe, kijungs}@kaist.ac.kr

Summary
- Goal: to detect anomalous hyperedges in a hyperedge stream
- Previous Work:
  - proposed algorithms for (pairwise) graphs
  - focused on only one of the aspects of anomalousness
- Proposed Method (HashNWalk):
  - an online algorithm for detecting anomalous hyperedges
  - detects structurally/temporally abnormal hyperedges
- Results:
  - Speed: processes each hyperedge in near real-time
  - Space: requires constant space, controlled by the user
  - Accuracy: outperforms the competitors up to 47% ↑ AUROC

Background: Hypergraphs
- Hypergraphs model group interactions
  - each hyperedge is a subset of any number of nodes
- In many real-world scenarios, hypergraphs evolve over time
  - a hyperedge stream \( \{(e_1, t_1), \ldots, (e_n, t_n)\} \) is a sequence of hyperedges

Background: Random Walk
- Random walk based on edge-dependent vertex weights for exploiting higher-order information
  - If the current node is \( u \),
    1. Select a hyperedge \( e \) that contains node \( u \) (i.e., \( u \in e \)) with probability proportional to the weight \( \omega(e) \).
    2. Select a node \( v \in e \) with probability proportional to the edge-dependent vertex weight \( \gamma(e)(v) \).
    3. Walk to node \( v \).

Problem Definition
- Anomalies in Hypergraphs:
  - Unexpected hyperedges consist of unnatural comb. of nodes
  - Bursty hyperedges appear in bursts in a short period of time
- Formal Problem Definition:
  - Given: a hyperedge stream
  - Detect: anomalous (i.e., unexpected/bursty) hyperedges
  - Desired: (a) in near real-time
    (b) using constant space

Proposed Algorithm: HashNWalk
- (1) Hypergraph Summarization
  - a new hyperedge arrives in the input hyperedge stream
  - nodes are merged into M supernodes by hashing
  - each hyperedge is represented as an \( M \)-dimensional vector
- (2) Incremental Update
  - the transition probability of supernode \( u \) → \( v \)
    - \( \bar{P}_{uv} \) is computed from \( S \) and \( T \)
    - They are incrementally updated in response to new hyperedges

Experimental Results
Q1. Performance: HashNWalk is accurate and fast in a real dataset (credit card transactions) and two semi-real datasets

Q2. Discoveries: HashNWalk detects meaningful events.
(1) Case study in DBLP hypergraph
(2) Case study in cite-patent hypergraph

Q3. Scalability: HashNWalk scales linearly with the hypergraph size
- Reproducibility: source code & datasets are available at:
  https://github.com/geonlee0325/HashNWalk