An Approach For Concept Drift Detection in a Graph Stream Using a Discriminative Subgraph

Ramesh Paudel and Dr. William Eberle
Department of Computer Science

Abstract
- With the emergence of the complex network (social media, sensor network, world wide web) the interest in graph mining has increased.
- In a streaming scenario, the concept to be learned might change over time.
- We propose a three-step approach to detect concept-drift detection on the graph streams:
  - **Subgraph Generation**: Find subgraphs (discriminative) for graphs in the stream.
  - **Entropy Calculation**: Measures distribution of current window in the graph stream by computing the entropy.
  - **Drift Detection**: Detect drift in the series of entropy values by moving one step forward in the sliding window.

Step 1 – Subgraph Generation
- Find discriminative subgraphs using the minimum description length (MDL) principle.
- Discriminative subgraph is the one that minimizes: 
  $$\text{MDL}(G, S) = \text{DL}(G) + \text{DL}(S)$$
where G is the entire graph, S is the subgraph, DL(G) is the description length of the subgraph, DL(S) is the description length of G after compressing it using S.

Step 2 – Entropy Calculation
- The probability of each discriminative subgraph $S_i$ in the current window with respect to the graph $G_i$ is:
  $$P(S_i | G_i) = \frac{\text{DL}(S_i)}{\text{DL}(G_i)}$$
- Entropy of the window based on subgraphs w.r.t. the graphs in W is defined as:
  $$\epsilon(W) = -\sum P(S_i) \log P(S_i | G_i)$$
where $P(S_i)$ is the fraction of subgraphs $S_i$ in W.

Step 3 – Change Detection
- Use direct density-ratio estimation approach called Relative Unconstrained Least-Squares Importance Fitting (RuLSIF) [1].
- The a-relative PE divergence ($D_{PE,a}$) gives the estimate of change.

Dataset and the Graph Structure

Algorithm

Methodology

Research Objective
- Design a state-of-the-art concept-drift detection method on graph streams.

Results

Experiment
- Each experiment is run 50 times by randomizing the graphs in the streams keeping the drift point the same.
- Using synthetic graph streams, the effect of the window size (|W|) is studied.
- Using the best W (|W| = 50), the performance of DDSD with GEM [3] and Zambon et al. [4] on 4 different real-world datasets is compared.
- For drift detection (step - 3), n = 50, k = 10, and α = 0.1 are used.

Conclusion
- Proposed a novel unsupervised algorithm for drift detection on graph streams called Discriminative Subgraph-based Drift Detection (DDSD).
- Performed several experiments on synthetic as well as real-world data.
- Outperformed both baseline approaches in terms of the DoD.
- Similar DDR and FA1000 with Zambon et al.
- In conclusion, the unsupervised nature of DDSD makes it a more robust in a streaming scenario.

Future Directions
- Investigate the scalability of our approach.
- Use our drift detection approach along with other learning models and investigate if the accuracy of the learning model can be improved by effectively detecting the drift.
- Test the performance against gradual drifts scenarios.

Acknowledgements
- Funding provided by Tennessee Tech University, College of Engineering for achieving Carnegie classification.

Reference