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.

Research Objective

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

Methodology

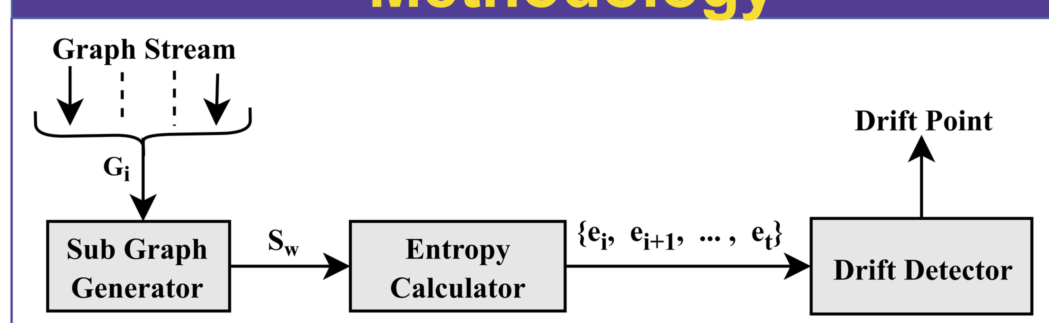

Fig 1. Methodology for Concept Drift Detection

Algorithm

Algorithm 1: Discriminative Subgraph-based Drift Detector (DSDD)

Input: Window Size = W
Graph Stream $G_s = \{G_1, G_2, \ldots, G_i, \ldots\}$
Output: Drift Points, False Alarm

Set $t = 0, n = 50, k = 10, \alpha = 0.1, fold = 5$
Initialize
$S_w = \{\}$: Holds subgraph and their count in G_t
Drift = []
$t_{buffer} \leftarrow (2n + k - 1)$
$i_{next} \leftarrow (2n + k - 1)$
while *not end of stream* **do**
  $t \leftarrow t + 1$
  $S_t \leftarrow$ `getDiscriminativeSubgraphs(`G_t`)`
  $S_w \leftarrow S_w \cup S_t$ `//update` S_w `with new subgraphs`
  if $t >= W$ **then**
    $i \leftarrow t - W$
    $e[i] \leftarrow$ `getWindowEntropy(`S_w`)  //using Eq. 2`
    if $i >= t_{buffer}$ *and* $i == i_{next}$ **then**
      $e_{cur} \leftarrow e[i - t_{buffer} : i]$
      $\hat{PE}_\alpha \leftarrow$ `getChangeScore(`e_{cur}, n, k, α`, fold)`
      if $\hat{PE}_\alpha > \eta$ **then**
        Append t in Drift
        $i_{next} \leftarrow i + (2n + k)$
      else
        $i_{next} \leftarrow i + 1$
      end
    end if
    $S_w \leftarrow S_w - S_{t-|w|}$
  end if
end

Step 1– Subgraph Generation

- Find discriminative subgraphs using the minimum description length (MDL) principle.
- Discriminative subgraph is the one that minimizes :

$$M(S, G) = DL(G|S) + DL(S)$$

where G is the entire graph, S is the subgraph, DL(S) is the description length of the subgraph, DL(G|S) is the description length of G after compressing it using S.

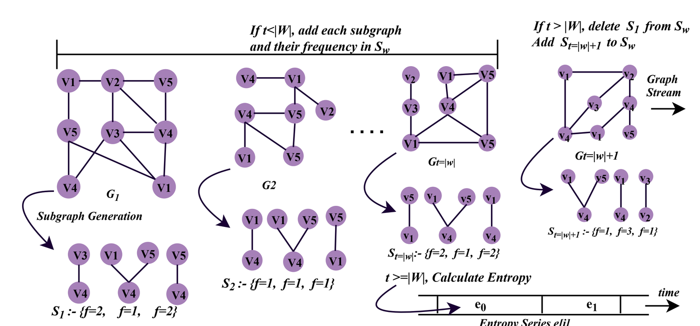

Fig 2. Illustration of discriminative subgraph generation from graph stream

Step 2 – Entropy Calculation

- The probability of each discriminative subgraph S_i in the current window with respect to the graph G_i is:

$$P(S_i|G_i) = \frac{r_{S_i}^{G_i}}{\sum_{j=1}^{|G|} r_{S_j}^{G_j}} \quad (1)$$

- Entropy of the window based on subgraphs w.r.t. the graphs in W is defined as:

$$e(W) = -\sum_{j=1}^{|S|} P(S_j) \sum_{i=1}^{N} P(S_i|G_i) \log_2 P(S_i|G_i) \quad (2)$$

where $P(S_i)$ *is the fraction of subgraphs* S_i *in* W

$P(S_i) = \frac{S_i}{\sum_{j=1}^{N} S_j}$, *N is the total number of subgraph in W*

Step 3 – Change Detection

- Use direct density-ratio estimation approach called Relative Unconstrained Least-Squares Importance Fitting (RuLSIF) [1].
- The α−relative PE divergence ($\widehat{PE}_\alpha$) gives the estimate of change.

$$\widehat{PE}_\alpha = -\frac{\alpha}{2n}\sum_{i=1}^{n} \hat{g}(Y_i)^2 - \frac{1-\alpha}{2n}\sum_{j=1}^{n} \hat{g}(Y'_j)^2 + \frac{1}{n}\sum_{i=1}^{n} \hat{g}(Y'_i) - \frac{1}{2}$$

where $\hat{g}(Y) = \sum_{l=1}^{n} \hat{\theta}_l \; K(Y, Y_l)$ and $K(Y, Y_l) = \exp\left(-\frac{\|Y - Y'\|^2}{2\sigma^2}\right)$

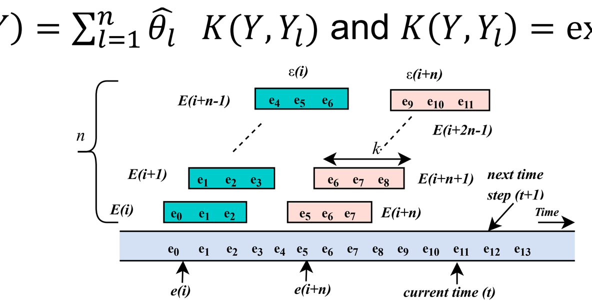

Fig 3. Change detection setup on entropy sequence

Dataset and the Graph Structure

Graph Stream Synthetic	Graph Stream Real World	# of Graph in Class A	# of Graph in Class B	Drift Points
SD1		1000	1000	1001
SD2		3000	3000	every 1001^{st} step
	DBLP	1000	1000	1001
	AIDS	1600	400	1601
	Muta	2401	1936	2402
	DoS Attack	2979	221	2980

Table 1. Total graph in graph stream dataset and the drift point

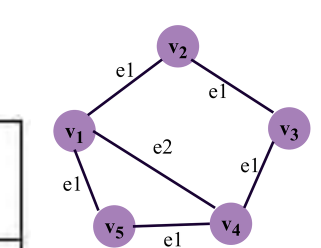

Fig 4. Positive and negative graph structure used in synthetic dataset

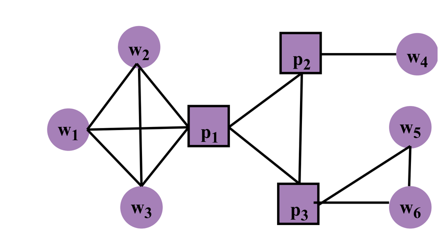

Fig 5. Graph structure of a DBLP citation network

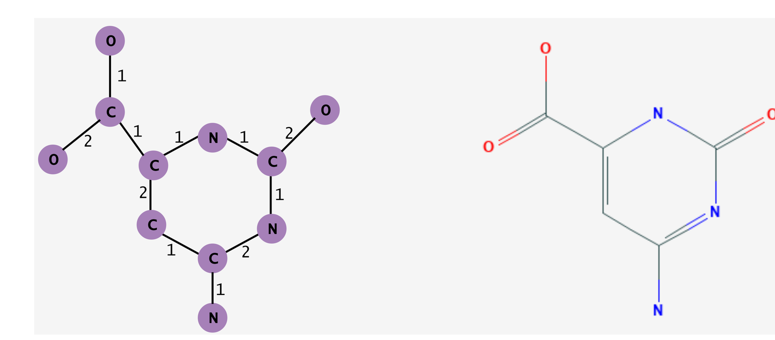

Fig 6. Chemical compound and its graph structure in IAM benchmark dataset

Results

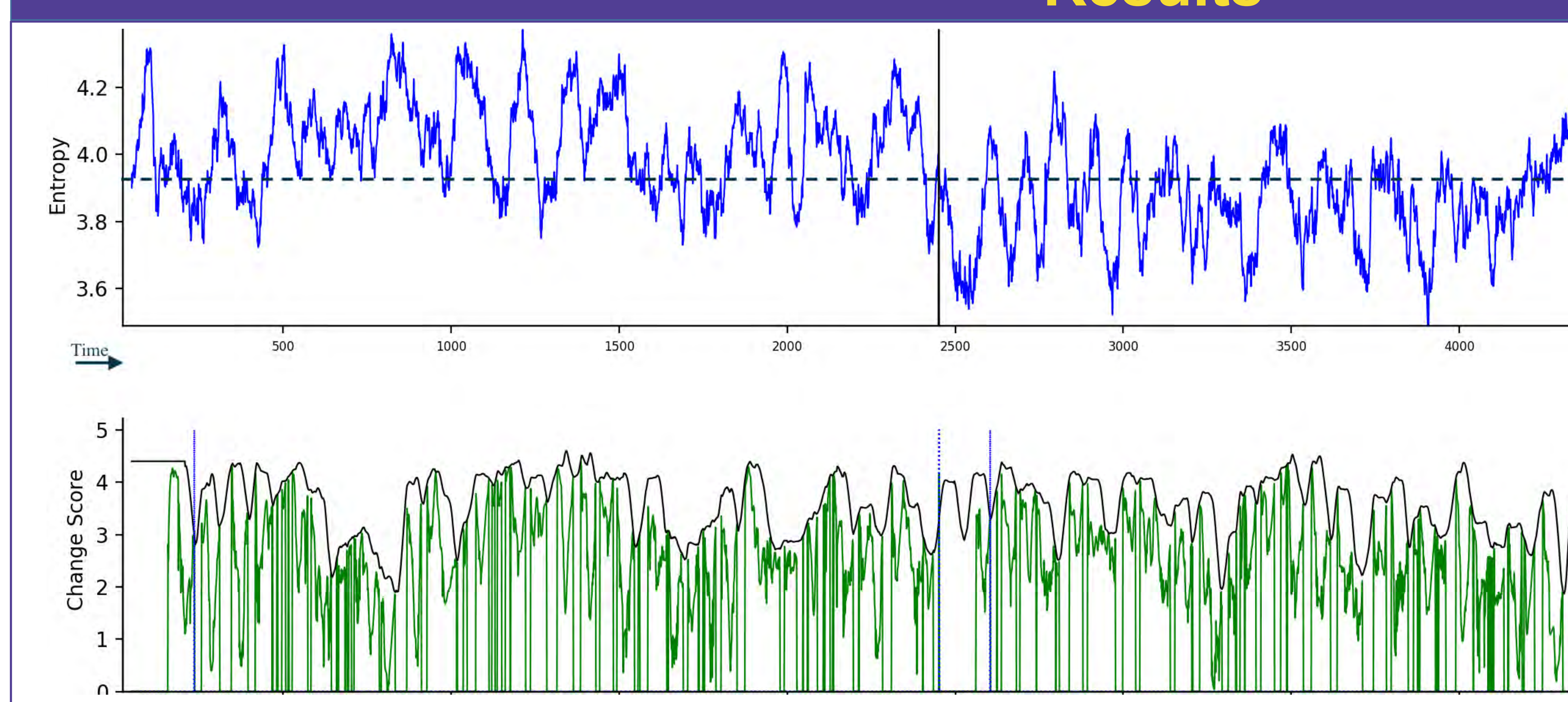

Fig 7. Entropy and change score on mutagenicity dataset
X-axis represents time. The vertical black lines in the entropy plot are the real drift point. The black plot on the change score plot is the threshold and vertical dotted blue lines are the detected drift points.

Methods	DDR μ	DDR σ	DoD μ	DoD σ	FA1000 μ	FA1000 σ
DBLP Citation Network						
DSDD	**1.0**	0	**19.24**	19.16	1.14	0.24
GEM	1.0	0	339	88.9	10.5	2.15
Zambon et. al.	1.0	0	437.5	-	**0.08**	0.08
AIDS						
DSDD	**1.0**	0	**17.48**	19.23	**1.11**	0.23
GEM	0	0	n/a	n/a	1.18	0.70
Zambon et. al.	1.0	0	62.5	-	1.31	0.45
Mutagenicity						
DSDD	**1.0**	0	**22.8**	20.63	0.60	0.18
GEM	0.46	0.50	124.12	161.0	1.37	0.42
Zambon et. al.	1.0	0	187.5	-	**0.29**	0.30
Network DoS Attack						
DSDD	**1.0**	0	21.16	21.24	0.81	0.25
GEM	1.0	0	**4.92**	1.63	11.05	2.98
Zambon et. al.	1.0	0	62.5	-	**0.065**	0.048

Table 2: Result obtained by DSDD and baseline methods on real-world datasets
DDR (Drift Detection Rate), DoD (Delay of Detection), FA1000 (False Anomalies per thousand time step)

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 DSDD 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 (DSDD)**.
- Performed several experiments on synthetic as well as real-world data.
- It outperformed both baseline approaches in term of the DoD.
- Similar DDR and FA1000 with Zambon et. al.
- In conclusion, the unsupervised nature of DSDD 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.

Reference

1. Song Liu, Makoto Yamada, Nigel Collier, and Masashi Sugiyama. 2013. Change point detection in time-series data by relative density-ratio estimation. Neural Networks 43 (2013), 72–83.
2. Diane J Cook and Lawrence B Holder. 1993. Substructure discovery using minimum description length and background knowledge. Journal of Artificial Intelligence Research 1 (1993), 231–255.
3. Yibo Yao and Lawrence B Holder. 2016. Detecting concept drift in classification over streaming graphs. In KDD Workshop on Mining and Learning with Graphs (MLG), San Francisco, CA. 2134–2142.
4. Daniele Zambon, Cesare Alippi, and Lorenzo Livi. 2018. Concept drift and anomaly detection in graph streams. IEEE transactions on neural networks and learning systems 29, 99 (2018), 1–14..

Acknowlegements

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