Anomaly Detection on Attributed Networks

Dylan Walker, Shengnan Miao, Bao Pham, Max Schwartz

Rensselaer Polytechnic Institute

December 5, 2021

Outline

Anomaly detection

- Anomaly Detection is the process of determining elements in a dataset that have a behavior that deviates from the rest of the dataset.
- Challenges remain for anomaly detection on attributed networks:
  (1) Network sparsity - the network structure could be very sparse on real-world attributed networks.
  (2) Data nonlinearity - the node interactions and nodal attributes are highly non-linear in nature while existing anomaly detectors mainly model the attributed networks with linear mechanisms.
  (3) Complex modality interactions - attributed networks usually have complex interactions for anomaly detection.

Attributed networks

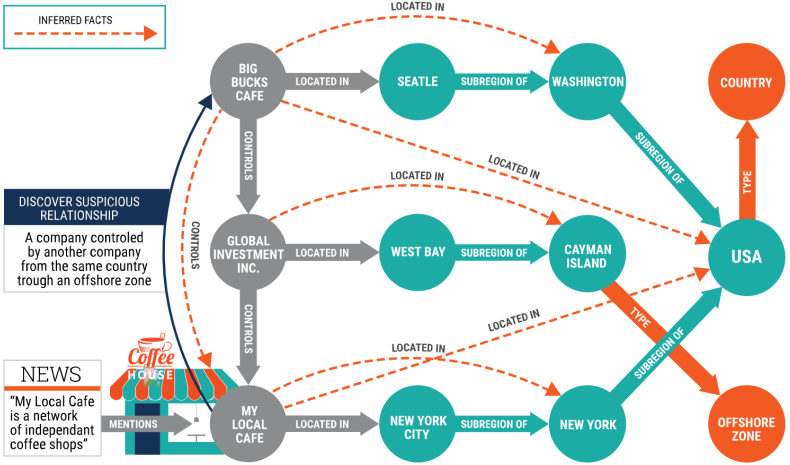
- An **attributed network** $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$ is an undirected graph with vertex set V, edge set E, and node feature matrix X.
- Conventionally, we let $N = |\mathcal{V}|$ denote the number of vertices and $m = |\mathcal{E}|$ denote the number of edges.
- Each node has a corresponding feature vector $x \in \mathbb{R}^k$, where k denotes the number of node features.
- Each edge in the network belongs to one of d different classes, where d denotes the number of distinct edge types.

Attributed networks (continued)

- The node feature matrix $X \in \mathbb{R}^{N \times k}$ compactly stores the node features for the entire network.
- The adjacency tensor $A \in \{0, 1\}^{d \times N \times N}$ stores an adjacency matrix for each of the d different edge types in the network.

Knowledge Graph

- A knowledge graph is a type of directed attributed network that models semantic data;
- Nodes represent real-world entities;
- Edges capture the relationships between entities.



Importance of Anomaly Detection in Knowledge Graphs

- Anomaly detection on knowledge graphs allows us to discover entities within a system that have suspicious behavior.
- Effective anomaly detection on large networks can be used in security efforts by highlighting the abnormal networks entities.
- For example, in a financial network anomaly detection can be applied to detect fraudulent accounts by analyzing their transaction patterns.

Problem statement

- Given an attributed network $\mathcal{G} = (V, E.X)$, our goal is to rank the vertices of $\mathcal{G}$ by how likely they are to be anomalous within the overall context of the network $\mathcal{G}$.
- The goal of our model is to learn a threshold value λ and a scoring function $f : v_i \to \mathbb{R}$ for each vertex $v_i \in V$ such that we can classify each node as anomalous or normal.
- Let y_i denote the output classification for node v_i under our model where $y_i = 1$ if v_i is anomalous and $y_i = 0$ if v_i is normal. Our goal is to learn f and λ such that:

$$y_i = \begin{cases} 1 & f(v_i) \geq \lambda \\ 0 & \text{otherwise} \end{cases}$$

Graph Convolutional Networks (GCN)

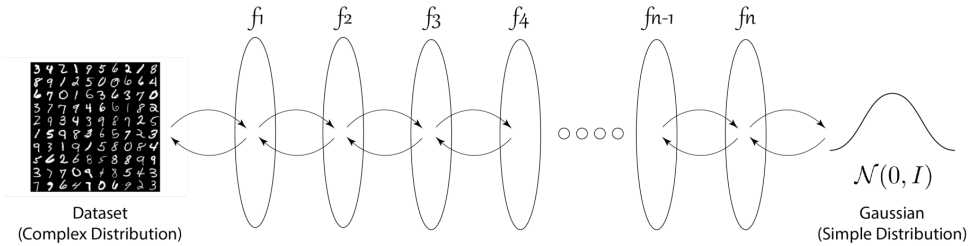
- Given an attributed network $\mathcal{G} = (V, E, X)$, we can use GCN's to learn embeddings $\{H^{(0)}, H^{(1)}, \ldots, H^{(L)}\}$

$$H^{(l+1)} = \sigma(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(l)}W^{(l)})$$

  with the following parameters:
  - $\hat{A} = A + I_N$ - Neighborhood adjacency matrix with self connections
  - $\hat{D} \in \mathbb{R}^{N \times N}$, the diagonal degree matrix of $\hat{A}$
  - $W^{(l)}$ - Weight matrix for layer l
  - σ - Nonlinear activation function
- Captures the critical inter-dependencies of network-structured data
- Node embeddings dependent on the local structure

Models and architectures

- Baseline: Auto-regressive Normalizing Flow Model:
  - An implementation based on graphAF
  - Normalizing flows is a generative modeling architecture that learns an invertible mapping from the data space to a latent probability space.
  - Auto-regressive normalizing flows learns a probability distribution that is used to sequentially reconstruct the network structure and node attributes



Graph Autoencoder Architecture

- Proposed Graph Autoencoder:
  - Preliminary
    In attributed network, we have a node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times k}$ and an adjacency matrix $\mathbf{A} \in \{0, 1\}^{d \times N \times N}$.
    Given an input dataset $(\mathbf{X}, \mathbf{A})$, the encoder $\mathrm{Enc}(\cdot)$, the decoder $\mathrm{Dec}(\cdot)$, then the learning process can be described as minimizing a cost function:

$$\min \mathbb{E}[\mathrm{dist}(\mathbf{X}, \mathrm{Dec}(\mathrm{Enc}(\mathbf{X}), \mathbf{A}, \mathrm{Dec}(\mathrm{Enc}(\mathbf{A}))]$$

    where $\mathrm{dist}(\cdot, \cdot)$ is a predefined distance metric.
  - Encoder
    A series of GCN layers are used to encode the graph neighborhoods into a latent embedding $\mathbf{Z}$.

Graph Autoencoder Architecture (continued)

- Structural Decoder
  The structural decoder learns an approximation of the adjacency tensor $\hat{\mathrm{A}}$
  $$\hat{\mathbf{A}} = S(\mathbf{Z}\mathbf{Z}^T)$$
  Where S is the element-wise sigmoid function, $S(x) = \frac{1}{1+e^{-x}}$
- Attribute Decoder
  The attribute decoder learns an approximation of the node feature matrix $\hat{\mathbf{X}}$
  $$\hat{\mathbf{X}} = GCN(\mathbf{A}, H^{(L)})$$
  where L is the depth of our graph convolutional networks (GCN).
- Loss Function
  $$\mathcal{L} = (1-\alpha)\|\mathbf{A} - \mathbf{A} \bullet \hat{\mathbf{A}}\|_F^2 + \alpha\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2$$
- Anomaly Scoring
  $$f(\mathbf{v}_i) = \mathcal{L}(\mathcal{N}(\mathbf{v}_i))$$

Anomaly detection for semantic network

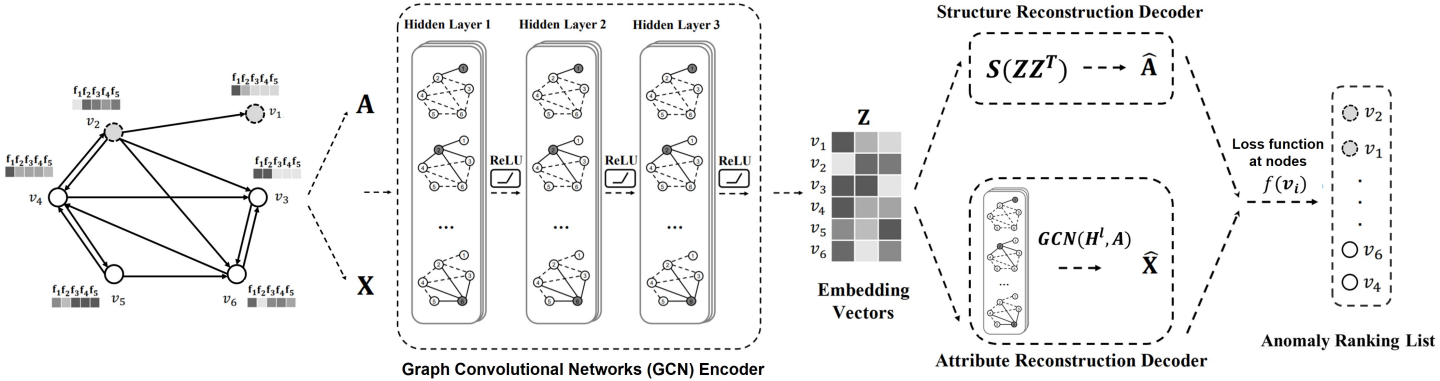


Figure 1: The overall framework of our proposed model for deep anomaly detection on semantic networks.

- Semantic networks are used in natural language processing applications such as semantic parsing and word-sense disambiguation.

Example of semantic network

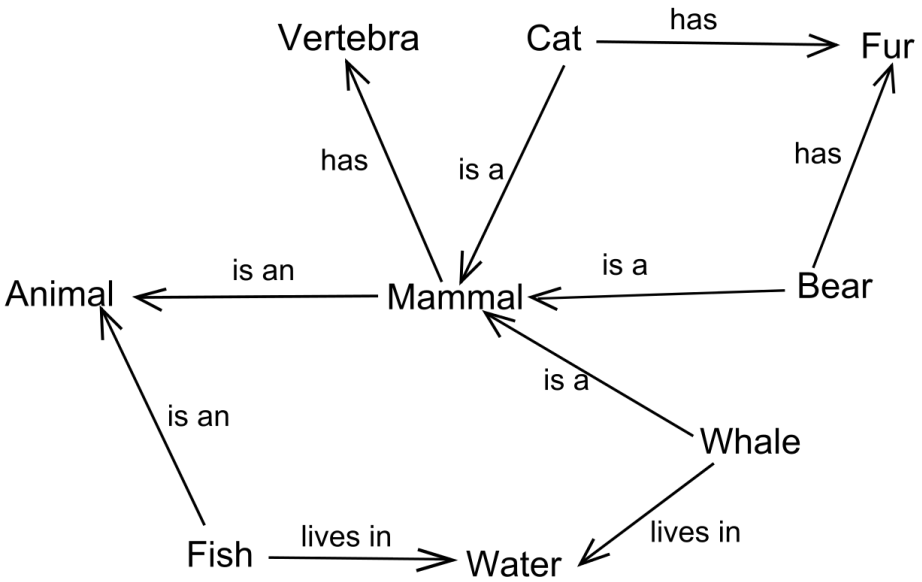


Figure 2: In this knowledge graph of semantic network, vertices represent concepts and edges represent semantic relations between concepts.

NELL Dataset

- NELL is a semantic network dataset built using the Never-Ending Language Learner
- The language learner reads from the web and produces a believed semantic network capturing the relationships between many different types of entities
- Each node is an individual entity and includes an attached query which we encode to form features for each node
- We trained with a subgraph containing 6182 nodes and 9649 edges over 60 different types of edge relations

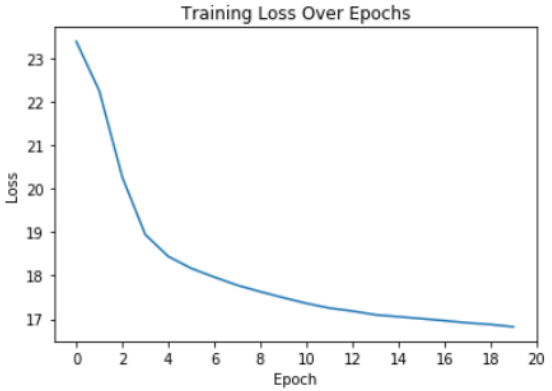
Train and Anomaly Data Formation

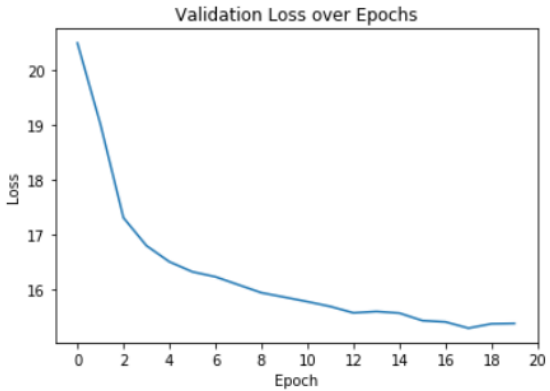
- To train and test our model we generated small sampled neighborhoods of the network as follows:
  - Randomly selected a node v
  - Use breadth-first search to find a local neighborhood
- To form a testing baseline, we introduce artificial anomalies into the network. We do this by introducing dense unexpected relationships into the network in the form of cliques.
  - Randomly select n nodes from the graph and form a clique amongst them.
  - Repeat m times to get a set of mn anomalous samples.

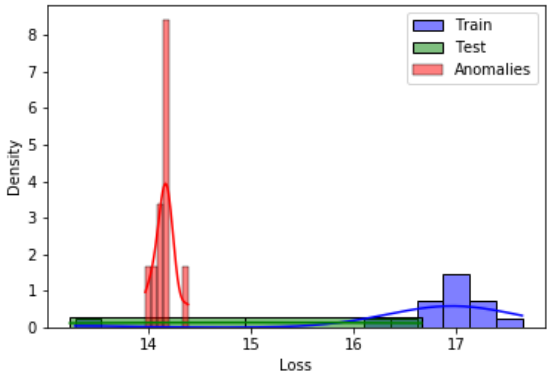
Numerical experiments

- Model is trained with
  - $m = 64$
  - $N_{train} = 1000$
  - $N_{val} = 100$
  - $N_{test} = 100$
  - $N_{anomaly} = 100$
- Evaluation Metrics
  - ROC-AUC
  - Precision
  - Recall
  - F1 score

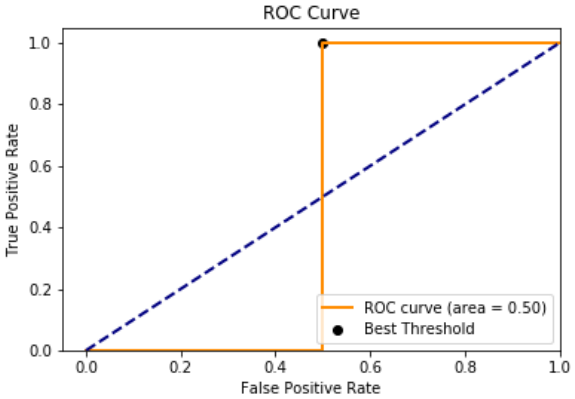
Results


Training Loss Over Epochs
Loss
Epoch
23
22
21
20
19
18
17
0
2
4
6
8
10
12
14
16
18
20


Validation Loss over Epochs
Loss
Epoch
20
19
18
17
16
0
2
4
6
8
10
12
14
16
18
20


Train
Test
Anomalies
Density
Loss
8
7
6
5
4
3
2
1
0
14
15
16
17

Results (continued)



Model	Precision-50	Precision-100	Recall-50	Recall-100	F1-score	ROC-AUC
Flow model						
Graph Autoencoder	0.90	0.90	0.90	0.90	0.90	0.50
% difference						

Note: Precision-50 and Recall-50 use 50 anomalies, all other metrics are in experiment with 100 anomalies.

Future work

- We can test and extend our anomaly detectors to apply on different network datasets, such as social networks, web-graph, product co-purchasing networks;
- We can use stochastic optimization and distributed learning to accelerate the training process and deal with large network datasets;
- We can investigate how robust our anomaly detector is in the presence of adversarial attacks and data poisoning attacks as intelligent attackers can inject malicious samples to avoid the anomalies being detected.

Reference

1. Yong, Z. X., Torrent, T. T. . Semi-supervised deep embedded clustering with anomaly detection for semantic frame induction. In *Proceedings of The 12th Language Resources and Evaluation Conference* (pp. 3509-3519).
2. Ding, K., Li, J., Bhanushali, R., Liu, H. . Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining* (pp. 594-602).
3. Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907.*
4. https://towardsdatascience.com/introduction-to-normalizing-flows-d002af262a4b
5. https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/