IT414 - Data Warehousing and Data Mining

Outlier detection using Unsupervised GraphSAGE

Team 9 Members

➢	Pratham Nayak	- 191IT241
➢	Aprameya Dash	- 191IT209
➢	Sarthak Jain	- 191IT145

Introduction

- Outlier detection refers to detection of objects that behave abnormally in comparison to other objects. Most existing outlier detection algorithms have difficulty detecting outliers that are mixed within normal object regions or around dense clusters.
- GNNs are neural networks that can be directly applied to graphs, and provide an easy way to do node-level, edge-level, and graph-level prediction tasks.
- GraphSAGE is a framework for inductive representation learning on large graphs. GraphSAGE is used to generate low-dimensional vector representations for nodes, and is especially useful for graphs that have rich node attribute information.
- This project develops a unsupervised learning based technique to perform outlier detection using GraphSAGE.

Some Additional Notes

Inductive representation learning is a type of machine learning approach that aims to learn representations of objects or data instances that generalize to new, unseen objects. In other words, it is the ability of a model to generate representations for objects that were not seen during training, but have similar properties to the objects in the training set.

This is in contrast to transductive representation learning, which aims to learn representations that are specific to the objects in the training set, and may not generalize well to new, unseen objects.

- The key idea behind GNNs is to recursively aggregate information from neighboring nodes in a graph, and use this information to update the representation of each node. This aggregation process is performed through a series of graph convolutional layers that operate on the node features and their connectivity patterns in the graph.

One of the benefits of batch normalization is that it helps to reduce the internal covariate shift, which is the change in the distribution of the input values to each layer of the network during training.

`optimizer.zero_grad()` is a method in PyTorch that sets the gradients of all parameters in the optimizer to zero. This is typically done before computing the gradients for a new batch of data during the training process of a neural network

Problem Statement

To identify the outliers using techniques based on unsupervised learning for a given a set of data points.

Datasets

S. No.	Dataset	No. of samples	No. of features	No. of outliers	Outlier Ratio
1	WBC	377	30	20	5.30 %
2	Wine	129	13	10	1.20 %
3	Pima	768	8	268	34.80 %
4	Glass	214	9	9	4.20 %
5	Vertebral	240	6	13	12.50 %
6	Ionosphere	351	33	127	36.10 %
7	Vowels	1452	12	46	3.10 %
8	letter	1600	32	100	6.25 %

Metrics

AUC Score

- It is the area under the receiver operating characteristic (ROC) curve.
- In the case of an outlier detection task, an ROC curve is drawn by plotting the FPR on the x-axis and TPR on the y-axis by varying the contamination level from 0 to 1.
- True Positive Rate (TPR) is the fraction of positive samples that are correctly classified.
- False Positive Rate (FPR) is the fraction of negative samples that are incorrectly classified.

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

Methodology

Graph Generation:

- Given a set of points X, we generate a graph where each node represents a point and for each node we add an edge between the node and its k nearest neighbours in terms of euclidean distance.
- Here k is a hyperparameter, we set k such that it is roughly equal to 20% of the data points.
- Euclidean distance between two points X_i and X_j is given by:

Learn From :

[OhMyGraphs: GraphSAGE and inductive representation learning | by Nabila Abraham | Analytics Vidhya | Medium]

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Methodology (Cont..)

Computing Node Embedding using GraphSAGE:

- GraphSAGE algorithm is used on graphs to capture graph structure. Let V be the set of all nodes in the graph, h^0_v represent the initial feature vector of the node v and N(v) be the set of neighbours of v, then the hidden vector representation of v in the lth layer is given by the following equations, where W_{self}, W_{neigh} and b are learnable parameters and σ is a non-linear activation.

$$h^l_{N(v)} = Aggregate(\{h^{l-1}_u, \forall u \in N(v)\})$$

$$h^l_v = \sigma(Concat(W^l_{self} \cdot h^{l-1}_v, W^l_{neigh} \cdot h^l_{N(v)}) + b^l)$$

Methodology (Cont..)

Training using Unsupervised Loss Function:

- To compute the embedding of a node, a GraphSAGE based GNN is constructed in which each layer's output is given by the equations given in previous slide.
- The GNN is trained with the objective to minimize the euclidean distance between the embeddings and the initial points.
- Since the number of outliers is small, the model will learn a mapping function that maps majority of the points (non-outliers) to an embedding that is close to the original data point, whereas the outliers will be mapped further away.

Methodology (Cont..)

Outlier Detection:

- After training of GNN, the embedding for every data point (or node) is calculated.
- Next, The euclidean distance between the data points and their embeddings is computed.
- Data points having higher euclidean distance with their embeddings are tagged as outliers.

Results

Dataset	K	LR	Dropout	Epochs	Batch Normalize	AUC Score	
						Proposed	Base Paper
wbc	20%	0.001	0.5	400	Yes	0.76	**0.98**
wine	20%	0.0005	0.5	90	No	**1.00**	1.00
pima	20%	0.005	0.5	50	Yes	0.65	**1.00**
glass	20%	0.0001	0.5	500	Yes	**0.95**	0.93
vertebral	20%	0.001	0.5	150	Yes	**0.78**	0.67
ionosphere	20%	0.0005	0.5	260	Yes	**0.99**	0.88
vowels	20%	0.001	0.5	500	Yes	**0.97**	0.93
letter	20%	0.001	0.5	400	Yes	**0.89**	0.89

THANK YOU