Oversampling in Heterogenous Graphs using SMOTE

Adhilsha A [*1] Deependra Singh [*1] Subhankar Mishra [2]

Abstract

Heterogeneous graphs serve as versatile representations of real-world data, making the challenge of imbalanced class distribution a pressing issue in machine learning. In this paper, we extend imbalanced learning techniques from homogeneous graphs to heterogeneous graph node classification. Inspired by the GraphSMOTE oversampling method for homogenous graphs, we introduce HeteroGSMOTE, tailored to heterogeneous graphs. We empirically evaluate HeteroGSMOTE across diverse heterogeneous graph datasets, demonstrating its applicability across domains and its superior performance compared to baseline methods. (some results represented). HeteroGSMOTE represents a significant advancement in addressing class imbalance in heterogeneous graph node classification, with implications for various real-world applications.

1. Introduction

Heterogeneous graphs represent a wide range of real-life data. Hence, downstream tasks such as node classification in heterogeneous graphs with imbalanced class distributions are a challenging problem in machine learning. While heterogeneous graph Neural Networks (HGNNs) have demonstrated remarkable performance in node classification tasks, they are often optimized for balanced class distributions. In many real-world applications, certain classes may have significantly fewer instances than others, leading to suboptimal performance when using traditional baseline models. To address this issue, we extend existing imbalanced learning techniques designed for homogenous graphs to heterogeneous graphs.

In this paper, we propose a novel oversampling approach in heterogeneous graphs called HeteroGSMOTE, which leverages Synthetic Minority Oversampling Techniques (SMOTE) on the representation matrices of the nodes to mitigate the class imbalance problem in heterogeneous graphs. SMOTE has proven effective in balancing class distributions in various domains, but its direct application to heterogeneous graph data presents unique challenges like requiring node type and meta-path-based contextual information.

To overcome these challenges, HeteroGSMOTE constructs a content-aggregated and neighbor-type-aggregated embedding space that encodes node similarities, facilitating the generation of synthetic samples that preserve the genuine relationships within the graph. Simultaneously, an edge generator for different node types or meta paths is trained to model the relational information between nodes, ensuring that the synthesized samples retain essential structural characteristics.

Our goal is to evaluate the effectiveness of GraphSMOTE on distinct heterogeneous graph datasets to show its application in various domains and try to demonstrate its superior performance compared to baseline methods.

2. Related Works

* GNNs (Zhao et al., 2021) * Homogenous and heterogeneous graphs and models (HetGNN(Zhang et al., 2019), HAN(Wang et al., 2019), MAgnn(Fu et al., 2020) and (Shi, 2022)) * class imbalance and GraphSMOTE Oversampling

3. Baseline Implementation

Our baseline models for implementing oversampling on graphs include GraphSMOTE (Zhao et al., 2021) and HeteroSMOTE leveraging GraphSAGE and HetGNN (Zhang et al., 2019) as baseline architecture for SMOTE.

3.1. Homogeneous graphs - GraphSMOTE

3.1.1. Dataset

We commenced our study with homogeneous graphs using the Cora dataset of 2708 paper nodes with 1433 dimensional attributes, 7 classes, and 5429 paper-paper citation edges, a well-established benchmark in the field. We employed the PyGeometric library to facilitate data loading and manipulation.

[*]Equal contribution [1]Department of Physical Sciences, NISER, Jatni, Odisha, 752050 [2]Department of Computer Sciences, NISER, Jatni, Odisha, 752050. Correspondence to: Adhilsha A <adhilsha.a@niser.ac.in>, Deependra Singh <deependra.singh@niser.ac.in>.

3.1.2. Model Architecture

The GraphSage model was chosen as our initial GNN architecture for the encoder and classifier. The encoder consists of 2 graph sage layers to produce the embedding for the nodes containing both attribute and relational information from the edge index tensor. The embedding is used to generate synthetic samples using smote and appended to the feature matrix. The new feature is passed through an edge predictor to generate the adjacency matrix for all the nodes. The edge predictor is trained on the original feature matrix. The final classification layer utilizes a GraphSAGE layer to produce the feature embedding of the synthetic nodes using their newly generated edges.

Synthetic Minority Oversampling Technique: SMOTE
In smote, for each node to be oversampled, the next closest node of the same class is found based on the Euclidean distance of their embedding, and their embedding is interpolated to produce a synthetic sample, which is given the same label as the parent nodes.

3.1.3. Implementation

From the node indices, we randomly selected indices for training, testing (55 for each class), and validation (25 for each class). For training, we selected 20 labels for the majority class and created an imbalance in the last three classes in the ratio in [1, 1, 1, 1, 0.8, 0.6, 0.4]. The learning rate was 0.0001, and the embedding dimension was taken 64. To establish a baseline for our experiments, we ran the model in two ways:
Method 1: Extracting the train, val, and test set and running on them only.
Method 2: On the complete dataset as per (Zhao et al., 2021) and training, testing on the masked indices only.

3.1.4. Results

The results of the implementation of GraphSMOTE are represented in the table below:

Method	Accuracy	Macro-Avg AUC-ROC	Macro Avg F1
1. No Smote	0.3636	0.7815	0.0051
1. With Smote	0.5481	0.8392	0.0641
2. No Smote	0.6494	0.9208	0.61347
2. With Smote	0.6545	0.9137	0.6007

Table 1. Results for GraphSMOTE

3.2. Heterogeneous graphs - HeteroSMOTE

To tackle class imbalance within heterogeneous graphs, we developed an alternative approach called HeteroSMOTE which incorporates GraphSMOTE method in HetGNN.

3.2.1. Dataset and Architecture

Shifting our focus to heterogeneous graphs, we turned to the A-miner A-II dataset, a rich source of academic information. It consists of labeled 28,645 author nodes in 4 classes, 21,044 paper nodes, and 18 venue nodes with 69,311 author-paper edges, 46,931 paper-paper edges, and 21,044 paper-venue edges. To handle the heterogeneity, we employed the HeteroData class within the PyGeometric library and implemented the HeteroGNN model.

3.2.2. Encoder

Content Aggregator
In this layer, the attribute matrix for each node is concatenated along the embedding dimension and then passed through a linear layer and nonlinearity to reduce its dimension back to the original embedding dimension, thereby combining the contextual information from all generated attributes of the nodes.

Type Based Neighbour Aggregator
Here, the content-based aggregated representation matrix for each node type is concatenated with the neighbor type-based aggregated feature matrix and then reduced using a linear layer. Firstly, for each neighbor, a list of frequently occurring neighbor nodes along different random walks is extracted. Top 10 frequently occurring author and paper nodes and top 3 venue nodes, thereby making a total of 9 matrices for each node type. For each node type, the three neighbor type-based aggregations are carried out, and they are combined using attention weights. This final aggregated matrix is concatenated with the node's content aggregated representation matrix and reduced using a linear layer following the GraphSAGE approach. Since we are classifying author nodes, the heterogeneous aggregation was carried out for venue nodes first (since they do not hold the direct edge with author nodes). Then, the paper nodes were aggregated using the updated venue nodes from the first step. Finally, the author node aggregation is carried out using the updated venue, paper node matrices, and its own content aggregated matrix.

3.2.3. Smote

The same procedure as GraphSMOTE is followed, and the new indices for the synthetic samples are appended to the training set.

3.2.4. Decoder

The decoder or the edge predictor is a neural network model that predicts the adjacency matrix for nodes of different types using a vanilla approach of matrix multiplication between the feature matrices of the two node types. A linear layer and nonlinearity are added to make the process learnable. The appended feature matrix after the smote is passed through the decoder, but the loss is calculated on the original slice of the feature matrix.

3.2.5. Classifier

Consists of a graph sage layer and a linear layer with nonlinearity. The graph sage layer can be replaced by a heterogeneous aggregator. It is used to get the new node embedding for all the authors nodes, old and synthetic, utilizing the predicted edges of the synthetic nodes for the same node type or using predicted edges for all types. The linear layer is used to reduce its dimension to 4, and subsequently, a softmax function is applied to each node to predict the class.

3.2.6. Implementation

Similar to our approach with homogeneous graphs, we introduced class imbalance in one of the 4 classes with a ratio of [1, 0.5, 1, 1] within the training classes. The majority class number is taken to be 200. We have chosen the same hyperparameters as the GraphSMOTE model. There are two losses for the model: an edge loss (L_e) for the decoder and a classification loss (L_c) for the classifier as CrossEntropyLoss. They are combined as such: $L = h \times L_e + L_c$, where h is a hyperparameter. There are three ways the model can be trained using the example of the graphSMOTE-:
1) Training all three- encoder, decoder, and classifier simultaneously by backward propagating combined losses from the classification and edge prediction task. (till Midway)
2) Pretraining the decoder beforehand.
3) Pretraining the decoder and encoder using the context loss as stated in (Zhang et al., 2019) and the edge prediction loss.

3.3. Results

The results are presented in the table and below for metrics on imbalanced data at 0.5 imbalance ratio for 100 epochs.
With Smote: Accuracy: 0.964, Macro AUC-ROC: 0.979, Macro F1: 0.430;
Without Smote- Accuracy: 0.945, Macro AUC-ROC: 0.977, Macro F1:0.458.

3.4. Conclusion and Future plans

Our findings demonstrate the effectiveness of GraphSMOTE for homogeneous graphs and the novel HeteroSMOTE approach for heterogeneous graphs. These experiments contribute to the advancement of GNN-based methods for academic paper classification, providing valuable insights into handling class imbalance in complex real-world datasets. In our future experiments, we intend to do the following. Run our model in different settings, like using LSTM instead of FC, using entire data instead of masking, and introducing

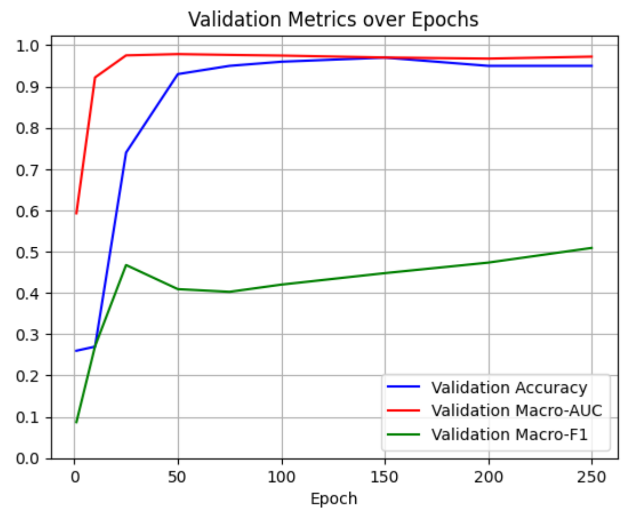

Figure 1. Validation Metrics vs Epoch Plot at imbalance ratio = 0.5

class imbalances in multiple classes within the dataset.We also plan to do a comparative study of our model with other baseline models. Additionally, we aspire to enrich our research by incorporating new and intriguing heterogeneous datasets, particularly those that provide labels for all nodes within the graph structure. This will enable us to explore the potential benefits of meta-path-based oversampling.

References

Fu, X., Zhang, J., Meng, Z., and King, I. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*, pp. 11. ACM, 2020. doi: 10.1145/3366423.3380297.

Shi, C. Heterogeneous graph neural networks. In Wu, L., Cui, P., Pei, J., and Zhao, L. (eds.), *Graph Neural Networks: Foundations, Frontiers, and Applications*, pp. 351–369. Springer Singapore, Singapore, 2022.

Wang, X., Ji, H., Shi, C., Wang, B., Cui, P., Yu, P., and Ye, Y. Heterogeneous graph attention network. In Sartor, J. B., D'Hondt, T., and De Meuter, W. (eds.), *Proceedings of WWW 2019*, pp. 4. ACM, 2019. doi: 10.475/123_4.

Zhang, C., Song, D., Huang, C., Swami, A., and Chawla, N. V. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 11. ACM, 2019. doi: 10.1145/3292500.3330961.

Zhao, T., Zhang, X., and Wang, S. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining*, pp. 9. ACM, 2021. doi: 10.1145/3437963.3441720.