Explicit Content Detection with Improved Random-walk Based Node Embedding Methods

Manqiu Liu
Georgia Institute of Technology
Atlanta, Georgia, USA
@gatech.edu

Sandeep N Sainath
Georgia Institute of Technology
Atlanta, Georgia, USA
ssainath3@gatech.edu

Qian Huang
Georgia Institute of Technology
Atlanta, Georgia, USA
qhuang301@gatech.edu

Jenna L Gottschalk
Georgia Institute of Technology
Atlanta, Georgia, USA
jgottschalk8@gatech.edu

ABSTRACT

Our project aims to increase predictive performance of classifying Twitch users who stream explicit content. Our dataset is an undirected graph containing 168,114 nodes and 6,797,557 edges, where nodes represent Twitch users and edges represent mutual follower relationships between them. The network has a density of 0.0005 and a transitivity of 0.0184. The goal of the project is to explore various node embedding algorithms to improve the performance of this node classification task. We consider RiWalk and MUSAE-EGO, two node embedding methods proposed in recent years and with promising strong performance. We hope to better classify explicit content and thus protect the underage users of Twitch.

1 INTRODUCTION

The primary objective of our project is to predict whether there is explicit content in Twitch streaming. We use the "Twitch Gamers Social Network" dataset [7], where nodes represent Twitch users and edges represent mutual follower relationships between them. Our task is to perform node-level classification on the network to predict whether a node is likely to stream explicit content.

One of the challenges in graph-based machine learning is creating embeddings that best capture the network information and structure. The original paper tests out several node embedding methods, including Walklets [5], a multiscale proximity-preserving node embedding, and Attributed Social Network Embedding (ASNE), a structural role-based node embedding algorithm [3]. When classifying whether a node is likely to stream explicit content, they achieve an AUC score of approximately 50-70%, and each suffers from various shortcomings. For example, Walklets lacks the ability to encode node attributes [6]. ASNE does not encode multi-scale or higher order relationships, and it does not scale well because explicit decomposition of the target matrix is more computationally intensive [6].

In attempts to improve predictive accuracy, we explore other advanced node embedding algorithms, specifically RiWalk and MUSAE-EGO. We then evaluate and compare their respective performance on a node classification task. Given the increasing popularity of video games and the growing participation of underage users, it is with great practical interest that we improve our ability to classify streaming content by its age-appropriateness.

2 LITERATURE SURVEY

Node embedding has been shown to be extremely beneficial in node classification tasks. In comparison to proximity preserved embedding methods such as SVD [1] brought up by Golub et al., random-walk based embedding methods such as node2vec [2] created by Grover et al. could explore more of the graph's latent structure, generating embeddings with structural information for downstream tasks. Classic random walk based embedding methods have several drawbacks such as the difficulty to choose proper hyperparameters and lack of awareness of local or global information. We researched two node embedding methods that tackled these shortcomings respectively as follows.

2.1 MUSAE EGO

In the original Twitch Gamers paper, Rozemberczki and Sarkar evaluated node embeddings that were either proximity preserving or structural role-based [7]. MUSAE-EGO [6] combines the benefits of these two approaches. MUSAE is a multi-scale structural role-based node embedding, and the EGO extension allows the model to learn local network information. This framework learns node and attribute embeddings based on local neighborhoods, such that nodes that are in neighborhoods of similar attributes have similar embeddings and attributes that are in similar neighborhoods have similar embeddings. By using attributed node embeddings that learn local feature information, MUSAE-EGO outperforms similar methods at predicting node attributes and can be used for transfer learning.

2.2 RiWalk

The RiWalk [4] created by Xuewei et al. is a structural embedding method, which has two procedures including role identification and network embedding. This method leverages graph kernel methods to perform role identification, then uses random walk to perform network embedding on subgraphs. RiWalk performs well on real-world datasets and outperforms state-of-the-art baselines on tasks such as node classification. Since RiWalk does not require heuristic feature engineering, it has a comparatively low computational cost, which is an advantage over other node embedding techniques. One shortcoming of RiWalk is that the embeddings are not task specific, as they are generated purely from random walk process without taking in any feature or tag information. We think it's possible that the random walk-based embeddings could have good performance in some tasks and bad performance in others. Besides, as they only perform random walks on subgraphs instead of the whole graph, we think RiWalk might miss long-run relationships among nodes.

3 DATA DESCRIPTION

Since we intend on measure performance solely based on various node embeddings and directly compare with the embedding methods used in Twitch Gamers paper [7], we use the same dataset as the one used in the original Twitch Gamers paper to replicate their experimental settings and data domain. The dataset, called Twitch Gamers Social Network, is pulled from the Stanford Network Analysis Project (SNAP), an open-source network analysis and graph mining library containing various code, documentation, and datasets pertaining to networks. The dataset represents a social network of Twitch users collected from the public Twitch API in Spring 2018. It consists of two CSV files containing edges between nodes and features of each node. The dataset is loaded using the NetworkX library in Python.

Nodes represent Twitch users and edges are undirected and represent mutual follower relationships between them. There are 168,114 nodes and 6,797,557 edges in total, and the entire network is a single strongly connected component (SCC). Nodes and edges do not have features associated with them, but each node also comes with labels containing information about categorical, numerical, and date information. Features include Views, Lifetime, Date Created, Date Updated, Dead Account, Affiliate Status, Broadcaster Language, and Explicit Content. The relevant categorical variables are Dead Account, Broadcaster Language, Affiliate Status, and Explicit Content, and Explicit Content has been chosen as the target variable. These categorical variables are binary except for Broadcaster Language which contains 20 language labels. Network density is 0.0005 and network transitivity is 0.0184. No additional pre-processing is performed on the node and edge data aside from implementing the baselines to retrieve node embeddings for our experiments. Additional pre-processing is performed on the node attributes for the implementation of the MUSAE-EGO node embeddings.

4 EXPERIMENTAL SETTINGS

We will use our Twitch dataset to perform a node classification task for evaluation. We choose logistic regression as our predictive model, with node embeddings as the input variable and a binary variable indicating whether this node streams explicit content as the target variable. Because this is a binary classification task, we will use mean macro-averaged AUC scores as our evaluation metrics, which will allow us to assess our classification power while also comparing it to our baseline paper. On our original data, we will perform train/test splits and vary our training data ratios from 0.1 to 0.9. Within each ratio, we will run ten train/test splits with different random seeds. System settings for the experiment involve running on 1 node on the Georgia Tech CoC-ICE cluster. This node has 10 cores with 5 GB RAM per core and a wall time of 8 hours.

5 METHODS

5.1 MUSAE EGO

MUSAE-EGO [6] created by Rozemberczki et al. is a structural role-based embedding algorithm that learns node and attribute embeddings based on random walks. Five walks are done per node, with a node sequence length of 80. Using a window size of 3, the walk iterates over the first 77 nodes of the sequence, which become source nodes, and the remaining nodes become target nodes. For each target node, a tuple of the source node and the target's features is added to a sub-corpus, and a tuple of the target node and the source's features is added to another sub-corpus. A Skip-gram optimizer with 5 negative samples is run on each sub-corpus to generate the individual embeddings that are concatenated to form multi-scale attributed node embeddings. A single epoch is run with a learning rate of 0.05. The statistics of node-feature pairs generated by the random walks estimate the joint probability of observing a feature before or after a node, which can be marginalized to estimate the stationary probability distributions of nodes and features. MUSAE node embeddings are represented by the following algorithm [6]:

$$log(cP^r FE^{-1}) - log(b) \text{ for } r = 1, ..., t$$

where A is the adjacency matrix, D is the diagonal degree matrix, F is the binary feature matrix; $c = \sum_{v,w} A_{v,w}$ is the volume of the graph; $P = D^{-1}A$ is the transition matrix of conditional probabilities; and $E = diag(1^T DF)$ is the diagonal matrix proportional to the probability of observing a node feature at the stationary distribution.

In the MUSAE-EGO formulation, an identity matrix I is appended to the feature matrix F, which adds a unique feature to each node that allows the embeddings to learn contextual proximity information. MUSAE-EGO node embeddings are represented by the following algorithm [6]:

$$log(\frac{c}{t}(\sum_{r=1}^{t} P^r)[F;I]E^{-1}) - log(b) \text{ for } r = 1, ..., t$$

The resulting embeddings are an implicit factorization of the node-feature pointwise mutual information matrix. We believe that MUSAE-EGO is well suited to our task because it combines the properties of proximity preserving and attributed node embeddings, which enable the model to excel at node attribute prediction. We use the author's own implementation of MUSAE-EGO, which can be found at https://github.com/benedekrozemberczki/MUSAE.

5.2 RiWalk

RiWalk [4] aims to encode nodes with similar local structures with similar embeddings. First, RiWalk creates subgraphs for each node. The subgraphs are created by including all the context nodes around the anchor nodes whose shortest paths to the anchor node are within a pre-defined radius. For all the subgraphs, RiWalk uses a shortest-path graph kernel to create a node relabeling process for all the context nodes. The labels will be further used in the random walks. This means that when creating embeddings using the random walks, the context nodes with the same labels will be considered as the same node even when they're in different subgraphs. After the role identification process, RiWalk uses a skip-gram model with negative sampling to treat node sequences as sentences, and then creates embeddings for each node. RiWalk achieves computational efficiency by performing random walks in parallel. The important hyperparameters of this method are number of dimensions of embeddings, number of walks per node, walk's length, window size for the skip-gram model, and radius to create subgraph. Based on the performance from the paper, we will set embedding dimension as 128, walk length as 10, number of walks as 30 considering the computational cost, window size as 10, radius as 4, number of epochs as 5. All other parameters will be kept as the default values [4]. This method could help create embeddings for this project, which could later be used in node classification task. As we have a relatively large dataset, computational efficiency of RiWalk could be extremely important in our project. We use the author's own implementation which can be found at https://github.com/maxuewei2/RiWalk.

6 EXPERIMENTS AND RESULTS

6.1 Twitch Paper Baselines

In the original paper, the node embedding methods Diff2Vec, DeepWalk, Walklets, RandNE, Role2Vec, ASNE, MUSAE, and FEATHER were used to classify nodes that stream explicit content. The performance on the classification task was measured by area under the curve scores on the test set. We adopt this evaluation metric to compare the performance of the baselines from the original paper to the node embedding techniques that we tested. The performance of baseline embeddings is summarized in the figure 1 and figure 2.

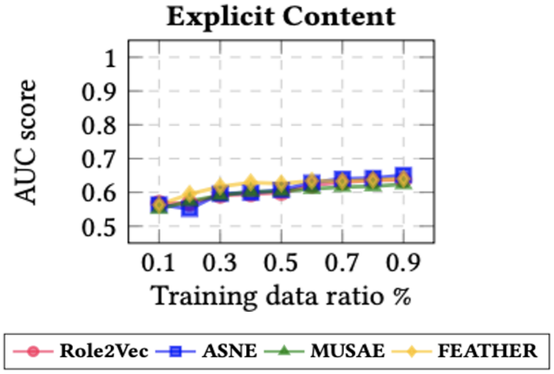

Figure 1: Proximity Preserving Methods Baseline Performance

These embeddings achieve a maximum AUC score of approximately 0.67, and Walklets achieves the highest performance of the node embeddings tested.

6.2 MUSAE EGO

Since MUSAE-EGO accepts a binary feature matrix as input, feature engineering was performed to convert node attributes to binary features. The attributes Views, Lifetime, Date Created, Date Updated, and Broadcaster Language were not originally encoded as binary features, and we converted all these features, except for Date Updated which had a similar value for most nodes, into binary features. Broadcaster Language was encoded using a one-hot encoding scheme, and a binary split at the median was used to encode Views, Date Created, and Lifetime. We created MUSAE and MUSAE-EGO embeddings using two different feature matrices. The first feature matrix included Dead Account, Affiliate Status, and Broadcaster Language. The second feature matrix included Dead Account, Affiliate Status, Broadcaster Language, Views, Lifetime, and Date Created. Although the original paper tests the performance of MUSAE node embeddings on classifying Explicit Content, no details are provided on the features selected for this method [6].

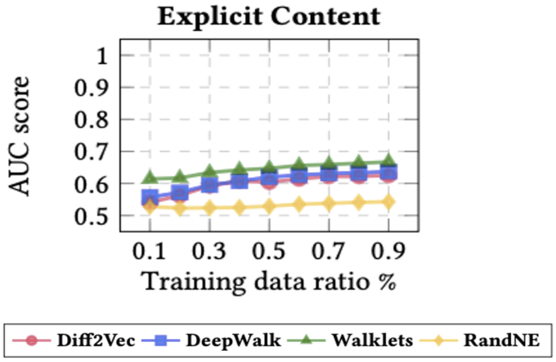

Figure 2: Structural Role-based Method Baseline Performance

The predictive performance of MUSAE and MUSAE-EGO node embedding techniques generated from the two different feature matrices on the classification task is summarized in the figure below.

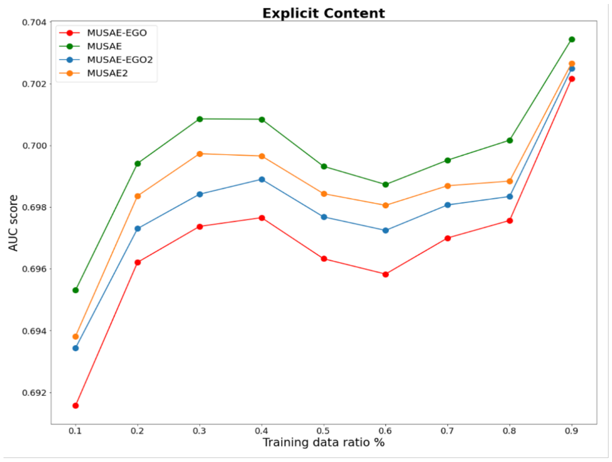

Figure 3: Predictive Performance of MUSAE and MUSAE-EGO

MUSAE is the best performing node embedding method. MUSAE embeddings generated from the first feature matrix achieve a maximum AUC score of 0.7034, and MUSAE embeddings generated from the second feature matrix achieve a maximum AUC score of 0.7027. This represents a 5% improvement from the MUSAE performance recorded in the original paper [6]. The original paper lacks details about feature engineering and hyperparameter tuning, and it is possible that the improved performance is due to differences in the attributes or hyperparameters chosen for the MUSAE embeddings.

Although MUSAE-EGO does not outperform MUSAE, the predictive performance of this node embedding technique outperforms the baselines explored in the original paper. MUSAE-EGO embeddings generated from the first feature matrix achieve a maximum AUC score of 0.7022, and MUSAE-EGO embeddings generated from the second feature matrix achieve a maximum AUC score of 0.7025.

We expected that MUSAE-EGO would achieve superior performance due to the additional network structure that is encoded in the embeddings. Rozemberczki et. al compared the performance of MUSAE and MUSAE-EGO node embeddings on node attribute classification, node attribute regression, and transfer learning tasks and found that MUSAE-EGO generally improves performance [6]. However, there were variable results for the node attribute classification tasks, with MUSAE-EGO achieving better performance on two datasets and MUSAE achieving better performance on other five datasets [6]. Although MUSAE-EGO embeddings provide additional contextual proximity information, encoding node attributes may be more important than encoding proximity to other nodes when using embeddings to predict whether a streamer uses explicit language. In fact, on the classification of Explicit Content using the Twitch Portugal dataset, MUSAE achieved an average micro F1 score of 0.672, while MUSAE-EGO achieved an average micro F1 of 0.671 [6].

6.3 RiWalk

Based on the embeddings generated from the RiWalk, we split the data into training set and testing set and performed logistic regression. The highest AUC score of RiWalk is about 0.57, which is slightly better than a random classifier. We also performed a sanity check on other labels using the embeddings and confirmed that RiWalk embeddings have worse performance in classifying explicit content compared to tasks such as classifying dead accounts.

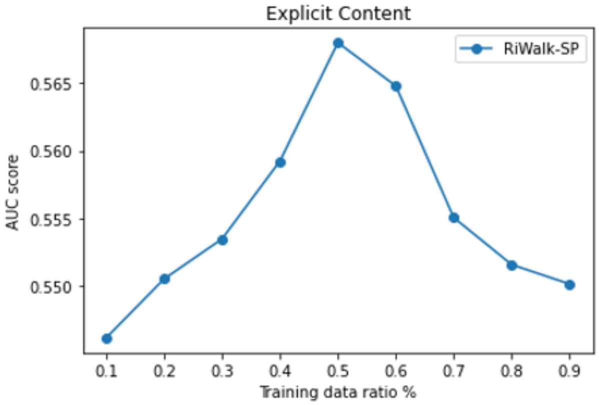

Figure 4: Predictive Performance of RiWalk

Comparing to the baselines from the dataset paper, we find that RiWalk has similar performance with RandNE, but performs worse than Diff2Vec, DeepWalk, and Walklets. Because of computational limitations, we only performed 30 walks for each node, which undermined the performance of the embeddings and could be the cause of RiWalk's overall suboptimal performance. Another possibility is that structural graph information does not capture whether a node will stream explicit content, which makes it harder for random walk based embedding methods to accurately predict it.

Comparing RiWalk with MUSAE, we get:

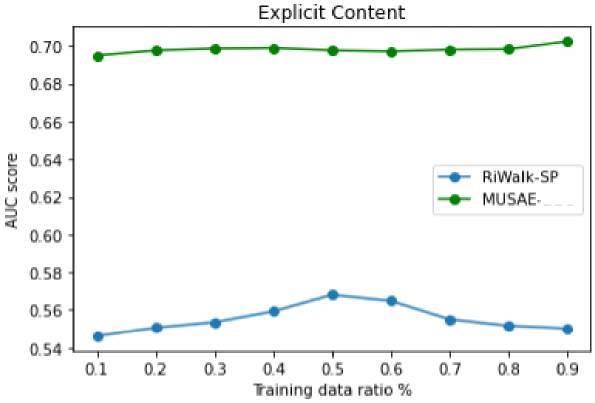

Figure 5: Predictive Performance of MUSAE and RiWalk

We can see that RiWalk has much lower performance than MUSAE.

7 CONCLUSION

The goal of the project is to explore various node embedding methods to improve the predictive accuracy of node classification, where we want to know whether a Twitch user is streaming explicit content or not. Of the two methods we explored, MUSAE and MUSAE-EGO outperform the baselines in the original Twitch Gamers paper, whereas RiWalk underperforms the baselines in the original paper, as well as the MUSAE and MUSAE-EGO node embeddings.

These two methods suffer shortcomings which could impair their performance on this classification task. Since MUSAE-EGO uses node attributes as input, feature engineering could alter the node embeddings created and impact the performance of the embeddings on classification tasks. A shortcoming of RiWalk is that even with parallel computing, the algorithm still requires many computational resources to achieve good performance. Due to the size of our dataset, we were only able to complete 30 walks per node with the available computational resources, which is not enough walks to take full advantage of the power of random walks. We believe that increasing walks per node would improve the embedding performance.

To improve the predictive performance of classifying explicit content, we could explore the following extensions on the methods that we implemented. For MUSAE-EGO, we can test the impact of feature engineering on the node embeddings and see if they outperform our existing implementations of MUSAE and MUSAE-EGO. For RiWalk, we can explore a wider range of hyperparameters with additional computational resources to see if different choices of hyperparameters result in improved predictive performance.

8 CONTRIBUTION

All team members have contributed a similar amount of effort.

REFERENCES

[1] Gene H. Golub and Christian Reinsch. 1971. Singular value decomposition and least squares solutions. *Linear Algebra* (1971), 134–151. http://people.duke.edu/~hpgavin/SystemID/References/Golub+Reinsch-NM-1970.pdf

[2] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[3] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2018. Attributed Social Network Embedding. *IEEE Trans. Knowl. Data Eng.* 30, 12 (2018), 2257–2270. https://doi.org/10.1109/TKDE.2018.2819980

[4] Xuewei Ma, Geng Qin, Zhiyang Qiu, Mingxin Zheng, and Zhe Wang. 2019. RiWalk: Fast Structural Node Embedding via Role Identification. *2019 IEEE International Conference on Data Mining (ICDM)* (2019), 478–487.

[5] Bryan Perozzi, Vivek Kulkarni, and Steven Skiena. 2016. Walklets: Multiscale Graph Embeddings for Interpretable Network Classification. *CoRR* abs/1605.02115 (2016). arXiv:1605.02115 http://arxiv.org/abs/1605.02115

[6] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2019. Multi-scale Attributed Node Embedding. *CoRR* abs/1909.13021 (2019). http://dblp.uni-trier.de/db/journals/corr/corr1909.html#abs-1909-13021

[7] Benedek Rozemberczki and Rik Sarkar. 2021. Twitch Gamers: a Dataset for Evaluating Proximity Preserving and Structural Role-based Node Embeddings. *CoRR* abs/2101.03091 (2021). arXiv:2101.03091 https://arxiv.org/abs/2101.03091