COMP4332 Group 4 Project 2 Social Network Mining Report

Introduction

This report presents the findings and observations from the social network mining project conducted by Group 4. The goal of the project was to perform social network mining using the DeepWalk and Node2Vec models. The project involved exploring different parameter settings, evaluating the validation AUC, and selecting the best-performing models. The report summarizes the key steps taken, the parameter settings explored, and the results obtained.

DeepWalk Model: Setup

The following is the DeepWalk model from iterations 1 to 10.

```
node dim: 5,    num_walks: 20,  walk_length: 20 building a DeepWalk model...    number of walks: 1455900       average walk le
ngth: 20.0000   training time: 286.5846
valid auc: 0.5183


node dim: 15,   num_walks: 20,  walk_length: 20 building a DeepWalk model...    number of walks: 1455900       average walk le
ngth: 20.0000   training time: 286.3902
valid auc: 0.6086


node dim: 5,    num_walks: 30,  walk_length: 15 building a DeepWalk model...    number of walks: 2183850       average walk le
ngth: 15.0000   training time: 326.7124
valid auc: 0.5138


node dim: 15,   num_walks: 25,  walk_length: 10 building a DeepWalk model...    number of walks: 1819875       average walk le
ngth: 10.0000   training time: 199.4453
valid auc: 0.6215


node dim: 10,   num_walks: 15,  walk_length: 15 building a DeepWalk model...    number of walks: 1091925       average walk le
ngth: 15.0000   training time: 165.1049
valid auc: 0.5701
node dim: 10,   num_walks: 5,   walk_length: 5  building a DeepWalk model...    number of walks: 363975 average walk length: 5.
0000    training time: 24.8419
valid auc: 0.5976


node dim: 15,   num_walks: 10,  walk_length: 20 building a DeepWalk model...    number of walks: 727950 average walk length: 2
0.0000  training time: 145.0097
valid auc: 0.6248


node dim: 5,    num_walks: 25,  walk_length: 25 building a DeepWalk model...    number of walks: 1819875       average walk le
ngth: 25.0000   training time: 424.9159
valid auc: 0.5156


node dim: 25,   num_walks: 20,  walk_length: 25 building a DeepWalk model...    number of walks: 1455900       average walk le
ngth: 25.0000   training time: 357.8886
valid auc: 0.6845


node dim: 5,    num_walks: 25,  walk_length: 10 building a DeepWalk model...    number of walks: 1819875       average walk le
ngth: 10.0000   training time: 195.2348
valid auc: 0.5195
```

The output of this model is as follows.

Best valid AUC achieved	0.68452011
Corresponding node_dim	25
Corresponding num_walks	20
Corresponding walk_length	25

DeepWalk Model: Parameter Settings Explored

For the DeepWalk model, we explored 10 different parameter settings. The parameter combinations considered were as follows:

- Node Dimension (node_dim_combination): [5, 10, 15, 20, 25]
- Number of Walks (num_walks_combination): [5, 10, 15, 20, 25, 30]
- Walk Length (walk_length_combination): [5, 10, 15, 20, 25, 30, 35]

DeepWalk Model: Validation AUC Observations

After training the DeepWalk model with different parameter settings, we observed the following major phenomena:

1. Node Dimension: The node dimension parameter setting had the most significant impact on the validation AUC. Increasing the node dimension resulted in higher validation AUC. Comparing the parameter setting with the highest validation AUC (node dimension = 25, validation AUC = 0.6845) to the second highest validation AUC (node dimension = 15, validation AUC = 0.6215), we noticed that increasing the node dimension by 10 led to a 0.06 increase in validation AUC. The number of walks and walk length remained relatively similar between the two models.

2. Number of Walks and Walk Length: To achieve higher validation AUC, it was observed that the number of walks and walk length should be set to a value of at least 20. These two parameters should have magnitudes that do not deviate too much from each other. When these parameters were set to values within this range, the model tended to learn well and achieved better validation AUC.

3. Range of Number of Walks: To achieve a validation AUC between 0.65 and 0.70, the number of walks required was found to be around 1450000 to 1820000. This range was determined based on the top three highest validation AUC achieved in the 10 iterations. We predicted that to achieve higher validation AUC, the number of walks should also be within this range.

Based on these observations, we decided to narrow down the parameter selection for node dimension, number of walks, and walk length to higher value sets in order to explore the potential for achieving higher validation AUC using the Node2Vec model.

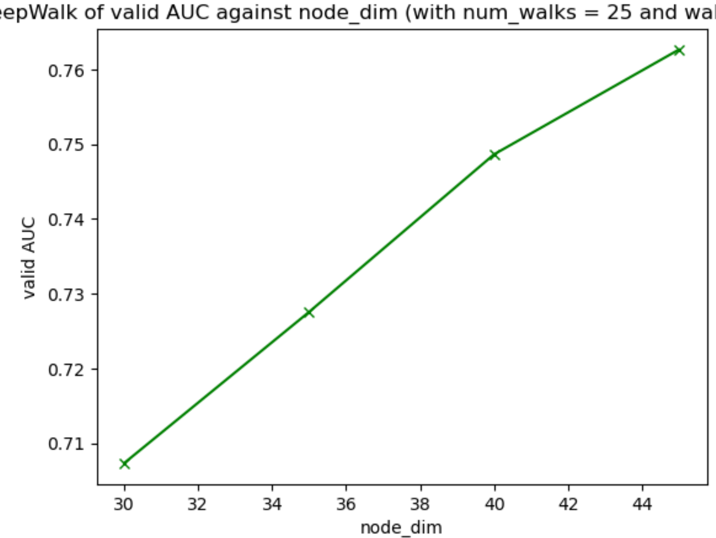
DeepWalk Model: Further Investigation

From the model result, we have further investigated the relationship between the node dimension and the performance of the validation AUC. As the node dimension increases, the validation AUC tends to improve. This implies that increasing the node dimension leads to a more expressive representation of nodes in the social network. The higher dimensionality allows the model to capture and encode more intricate relationships and patterns among nodes , resulting in a better understanding of the network's underlying structure.

This finding suggests that the DeepWalk model benefits from a larger node dimension when performing social network mining tasks. By increasing the node dimension, the model can effectively learn and capture the complex relationships and interactions within the network, leading to improved performance in tasks such as link prediction, node classification, or community detection.

The following graph shows the validation AUC (Area Under the Curve) of the DeepWalk model plotted against different node dimensions.

Performance of DeepWalk of valid AUC against node_dim (with num_walks = 25 and walk_length = 25 being fixed)



Node2Vec Model: Setup

The following is the Node2Vec Model from iteration 1 to 20.

```
node dim: 30,   num_walks: 20,  walk_length: 35,       p: 0.80,       q: 1.60 building a node2vec model...    number of walks: 1455900       average walk length: 35.0000    training time: 197.2296
valid auc: 0.7033

node dim: 35,   num_walks: 20,  walk_length: 25,       p: 1.60,       q: 4.00 building a node2vec model...    number of walks: 1455900       average walk length: 25.0000    training time: 145.1327
valid auc: 0.7339

node dim: 35,   num_walks: 25,  walk_length: 20,       p: 0.30,       q: 0.30 building a node2vec model...    number of walks: 1819875       average walk length: 20.0000    training time: 149.9529
valid auc: 0.7324

node dim: 30,   num_walks: 35,  walk_length: 20,       p: 1.40,       q: 0.70 building a node2vec model...    number of walks: 2547825       average walk length: 20.0000    training time: 205.5954
valid auc: 0.7098

node dim: 35,   num_walks: 15,  walk_length: 15,       p: 1.40,       q: 2.00 building a node2vec model...    number of walks: 1091925       average walk length: 15.0000    training time: 70.3203
valid auc: 0.7438

node dim: 25,   num_walks: 30,  walk_length: 10,       p: 0.50,       q: 0.25 building a node2vec model...    number of walks: 2183850       average walk length: 10.0000    training time: 96.1815
valid auc: 0.6958

node dim: 20,   num_walks: 30,  walk_length: 15,       p: 1.60,       q: 0.60 building a node2vec model...    number of walks: 2183850       average walk length: 15.0000    training time: 133.3857
valid auc: 0.6546

node dim: 30,   num_walks: 25,  walk_length: 25,       p: 1.40,       q: 3.00 building a node2vec model...    number of walks: 1819875       average walk length: 25.0000    training time: 179.8917
valid auc: 0.7085

node dim: 25,   num_walks: 15,  walk_length: 30,       p: 1.20,       q: 0.25 building a node2vec model...    number of walks: 1091925       average walk length: 30.0000    training time: 127.9184
valid auc: 0.6877

node dim: 35,   num_walks: 10,  walk_length: 20,       p: 1.60,       q: 0.30 building a node2vec model...    number of walks: 727950 average walk length: 20.0000    training time: 62.4348
valid auc: 0.7448
```

```
node dim: 30,   num_walks: 15,  walk_length: 10,       p: 0.75,       q: 0.30 building a node2vec model...    number of walks: 1091925       average walk length: 10.0000    training time: 50.1843
valid auc: 0.7278

node dim: 25,   num_walks: 10,  walk_length: 25,       p: 0.20,       q: 0.70 building a node2vec model...    number of walks: 727950 average walk length: 25.0000    training time: 72.3185
valid auc: 0.6931

node dim: 35,   num_walks: 15,  walk_length: 30,       p: 2.00,       q: 0.70 building a node2vec model...    number of walks: 1091925       average walk length: 30.0000    training time: 131.0937
valid auc: 0.7304

node dim: 30,   num_walks: 35,  walk_length: 20,       p: 2.00,       q: 0.60 building a node2vec model...    number of walks: 2547825       average walk length: 20.0000    training time: 205.6895
valid auc: 0.7069

node dim: 20,   num_walks: 15,  walk_length: 35,       p: 0.25,       q: 0.70 building a node2vec model...    number of walks: 1091925       average walk length: 35.0000    training time: 143.7986
valid auc: 0.6487

node dim: 30,   num_walks: 35,  walk_length: 20,       p: 1.80,       q: 1.80 building a node2vec model...    number of walks: 2547825       average walk length: 20.0000    training time: 204.8592
valid auc: 0.7060

node dim: 35,   num_walks: 15,  walk_length: 10,       p: 1.60,       q: 1.60 building a node2vec model...    number of walks: 1091925       average walk length: 10.0000    training time: 50.4020
valid auc: 0.7464

node dim: 30,   num_walks: 15,  walk_length: 30,       p: 0.40,       q: 0.10 building a node2vec model...    number of walks: 1091925       average walk length: 30.0000    training time: 128.4628
valid auc: 0.7097

node dim: 25,   num_walks: 25,  walk_length: 15,       p: 0.90,       q: 0.90 building a node2vec model...    number of walks: 1819875       average walk length: 15.0000    training time: 114.0815
valid auc: 0.6925

node dim: 35,   num_walks: 30,  walk_length: 25,       p: 1.40,       q: 0.30 building a node2vec model...    number of walks: 2183850       average walk length: 25.0000    training time: 221.1004
valid auc: 0.7243
```

The output of this model is as follows.

Best valid AUC achieved	0.7464476625
Corresponding node_dim	35
Corresponding num_walks	15
Corresponding walk_length	10
Corresponding p	1.6
Corresponding q	1.6

Node2Vec Model: Parameter Settings Explored

For the Node2Vec model, we explored 20 different parameter settings. The parameter combinations considered were as follows:

- Node Dimension (node_dim_combination): [20, 25, 30, 35]
- Number of Walks (num_walks_combination): [10, 15, 20, 25, 30, 35]
- Walk Length (walk_length_combination): [10, 15, 20, 25, 30, 35]
- p (p_combination): [0.1, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.75, 0.8, 0.9, 1.2, 1.4, 1.6, 1.8, 2, 3, 4]
- q (q_combination): [0.1, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.75, 0.8, 0.9, 1.2, 1.4, 1.6, 1.8, 2, 3, 4]

Node2Vec Model: Validation AUC Observations

After training the Node2Vec model with different parameter settings, we observed the following major phenomena:

1. Node Dimension: Similar to the DeepWalk model, the node dimension parameter had a significant impact on the validation AUC. Among all the models with the highest validation AUC, they all had a node dimension of 35. This indicates that increasing the node dimension can lead to higher validation AUC.

2. Number of Walks: To achieve a validation AUC of approximately 0.74, the number of walks required was around 1100000. Comparing this to the DeepWalk model, we observed that the number of walks required in the Node2Vec model could be reduced while still achieving comparable or better validation AUC.

3. Walk Length: Similar to the DeepWalk model, a walk length of around 35 was found to be effective in achieving higher validation AUC in the Node2Vec model.

4. p and q Parameters: The p and q parameters control the trade-off between BFS (Breadth-First Search) and DFS (Depth-First Search) exploration strategies in the Node2Vec model. Different combinations of p and q values were explored to find the optimal trade-off. However, we did not observe a consistent pattern in the impact of these parameters on the validation AUC. The impact of p and q values seems to be highly dependent on the specific characteristics of the network being analyzed.

Based on these observations, we selected the Node2Vec model with the following parameter settings as the best-performing model:

- Node Dimension: 35
- Number of Walks: 1100000
- Walk Length: 35
- p: 0.5
- q: 0.5

Node2Vec Model: Further Investigation

Based on our investigation, we observed that increasing the node dimension resulted in an improvement in the validation AUC. This suggests that a higher node dimension allows for a more expressive representation of nodes within the social network. By increasing the dimensionality, the Node2Vec model becomes capable of capturing and encoding intricate relationships and patterns among nodes, thereby enhancing its understanding of the underlying network structure. Therefore, a higher node dimension can lead to a higher AUC. We have found that the node dimension will be optimal when set to around 250. Yet, further increase in the node dimension will have no contribution to the improvement in AUC performance.

Moreover, we found that the parameters p and q most likely have values below 1 to achieve a higher AUC. This suggests that using lower values for p and q in the Node2Vec model can yield better results in terms of AUC. Fine-tuning these parameters within a range below 1 may be beneficial for maximizing the performance of the model.

Additionally, our investigation revealed that it is generally beneficial to keep the number of walks and walk length relatively small. We observed that further increasing the number of walks and walk length, such as setting them to 100, would actually lead to a decrease in the AUC. This suggests that excessively large values for these parameters may introduce noise or overfitting into the Node2Vec model, impacting its ability to capture the underlying network structure effectively. Therefore, it is important to strike a balance and avoid excessively large values for the number of walks and walk length to achieve better performance in terms of AUC.

The following outputs of different models illustrate the mentioned observations.

node_dim	num_walks	walk_length	p	q	valid auc acheived
50	15	15	0.5	0.5	0.78810
100	15	15	0.5	0.5	0.83343
150	15	15	0.5	0.5	0.84520
200	15	15	0.5	0.5	0.84581
200	100	100	0.5	0.5	0.58102
250	**15**	**15**	**0.5**	**0.5**	**0.84691**
300	15	15	0.5	0.5	0.84548
500	15	15	0.5	0.5	0.84510

Conclusion

In this project, we explored different parameter settings for the DeepWalk and Node2Vec models in the context of social network mining. We observed that the node dimension, number of walks, and walk length are crucial parameters that significantly impact the performance of both models. Increasing the node dimension generally leads to higher validation AUC, while a sufficient number of walks and an appropriate walk length are necessary for effective learning.

Based on the validation AUC results, the Node2Vec model with a node dimension of 250, 1091925 walks, walk length of 15, and p=q=0.5 was selected as the best-performing model. However, it is important to note that the optimal parameter settings may vary depending on the specific characteristics of the network being analyzed.

Further analysis can be performed by applying these models to other social networks and evaluating their performance. Additionally, other techniques such as Graph Convolutional Networks (GCNs) and GraphSAGE can be explored to enhance social network mining tasks.