Project Weave 2.0: Using Graph Neural Network for Startup Success Prediction

Xun Hu[1], Yigit Ihlamur[2]

[1]University of Oxford, UK
[2]Vela Partners, US

Abstract

Venture technology (Ventech) is an innovative way of investing in startups through the implementation of machine learning models. In this report, we use Graph Neural Network (GNN) to distinguish between successful and unsuccessful startups based on various attributes such as founders' work experiences and educational backgrounds. These companies are interconnected through different means such as being invested by the same investor, forming a large and complicated network. Therefore, graph neural networks become a suitable tool to analyse such data structure.

1 Introduction

Our approach to predict the success of a startup is inspired by recent research on node classification in graph structures. Let G = (V, E) be a graph, where V is the set of vertices or nodes and E is the set of edges. In the case of a startup network, nodes represent startups while edges represent relationships between startups. We can choose the relationship to be whether they have a common investor for reasons explained in the next paragraph.

Currently, GNN contains message passing layers, such as Graph Convolutional Network (GCN)[1] and Graph Attention Network (GAT)[2], which exploit the property of permutation equivariance. It updates the representation of each node by aggregating the messages from its immediate neighbours. Message passing layers update the representation of each node by aggregating the messages from its immediate neighbours, meaning a startup will aggregate messages from connected startups. Intuitively, a startup tends to share attributes with other startups invested by the same investor, either because each investor selects startups based on the same set of criteria, or startups will share the same resources from their investors. In transductive learning, we can include startups with and without labels in the same graph structure, and then apply GNN over the graph. The node feature vectors of those unlabelled nodes after training can be used to predict the success of a startup.

2 Data Preprocessing & Feature Engineering

The data used in this report is a subset of Moneyball 2.0 database, a propriety database of Vela Partners. It contains information of successful and unsuccessful startups and their enriched founder profiles through the use of web scraping methods.

2.1 Data Sets

Moneyball 2.0 database This database contains companies information and their founders' Linkedin URLs for both successful and unsuccessful companies. We use information from the database as shown below:

- org_uuid: Unique key of the company
- org_name: Name of the company
- founder_linkedin_url: Linkedin profile links of the company founders
- investment_type: Specifies if the funding round is angel, pre-seed, seed or series a
- raised_amount_usd: The total investment amount raised in US dollars
- investor_count: Number of investors
- investor_name: Names of investors
- investor_uuids: Unique keys of investors

Founder Profile Data This database contains founders' URLs and JSON strings storing their personal information as follows:

- founder_linkedin_url: Linkedin profile link of the founder
- founder_age: Age of the founder
- founder_education: List of academic institutions that the founder attended
- founder_degree: List of degrees completed
- founder_past_companies: List of past companies that the founder worked at
- founder_past_roles: List of past roles undertaken in previous companies

2.2 Feature Engineering

Since each company has multiple founders, and each founder has multiple educations and past work experience. For simplicity, we only consider the main founder of the company and the latest education and job for each founder. As such, each startup has 7 attributes: founder_education, founder_degree, founder_past_company, founder_past_role, founder_age, raised_amount_usd, investor_count.

The first four attributes are all text strings which might contain synonyms such as "Co-founder CEO" versus "Founder Chief Executive Officer". In order to avoid confusion, we use OpenAI's embedding model[1], *text-embedding-ada-002*, to convert texts into 1536-dimensional dense vectors, which can be used as part of feature vectors.

The last three attributes are numbers, but the scale is different. In order to re-scale these values, we use standardisation:

$$\bar{x} = \frac{x - \mu}{\sigma} \tag{1}$$

where x is the value of the attribute, μ is its mean, and σ is its standard deviation.

Finally, we can concatenate all these features together to form a 6147-dimensional feature vector for each node.

3 Methodology

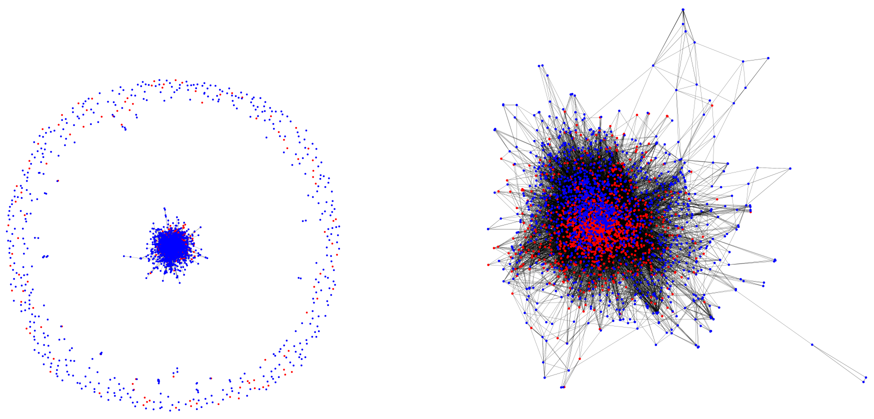
3.1 Finding Edges

Right now we have obtained feature vectors for all companies. Labels can be inferred directly for each company: 1 for successful companies, and 0 for unsuccessful companies. However, we still need to find edges between companies. We can create a dictionary with investor_uuids as keys and org_uuids as values. An edge can be created for any pair of companies under the same investor.

3.2 Graph Structure

To store and visualise our data, we use *NetworkX*, a Python package for the creation and study of complex networks. We can create a node for each company and store its feature vector and label in the node. Edges are also created from the above dictionary. Visualisations of graphs are shown in Figure 1.

[1]https://platform.openai.com/docs/guides/embeddings/what-are-embeddings



(a) All Companies (b) Connected Companie

Figure 1: graph structure for companies
Red: successful; Blue: unsuccessful

3.3 Reducing floating companies and bottlenecks

Literature review shows that bottlenecks in graph neural networks can cause pathological behaviours such as over-squashing [3], where information stored in the nodes will increase drastically with model depth. In order to reduce the number of bottlenecks, we will remove companies with fewer than three edges, including those companies floating around with no connection to other companies that we know of. The rationale behind this is that it would be extremely risky to invest in companies that we lack information on. The graph we obtained after a few iterations of removing less connected graphs is shown in Figure 1b.

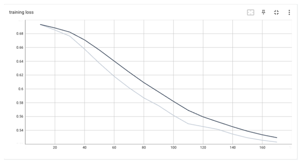
3.4 Applying Graph Neural Networks to Graphs

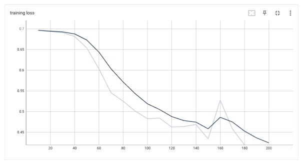
We experimented with three neural networks, namely, Graph Convolutional Network (GCN), Graph Attention Network (GAT), and Multi-layer Perceptron (MLP) [4], where MLP is used as a baseline model and it does not use edge information. We use Adam optimiser with an initial learning rate of 0.01 and weight decay of 5×10^{-4}. The number of epochs required to obtain a decent result is 180 for GCN, 200 for GAT, and 150 for MLP, respectively. Metrics used to evaluate the performance of a model are accuracy and precision.

$$\text{Precision} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

where P: positive, TP: true positive, FN: false negative

4 Results & Discussion


(a) GCN


(b) GAT

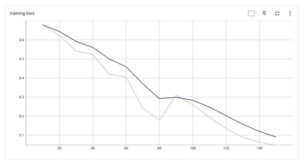

(c) MLP

Figure 2: Training losses for three models

The results obtained for three machine learning models are shown in table 1, and training losses are shown in Fig 2. It is clear that both GCN and GAT perform better than the baseline model, MLP, by a significant margin, because GCN and GAT are designed for graph structure while MLP ignores the node connections and only uses feature vectors to classify nodes. In our experiment, GCN performs slightly better than GAT in terms of precision and slightly worse in terms of accuracy, but these differences are minor. Surprisingly, precision is much higher than accuracy for all models in our experiment, which is good for the purpose of investing in startups. It means that the return on investment is likely to be even higher than expected.

Table 1: Machine Learning Model Results

Model	**Accuracy**	**Precision**
GCN	0.728	0.849
GAT	0.738	0.832
MLP	0.671	0.724

As such, Project Weave 2.0 shows the plausibility of using Graph Neural Networks to predict the success rate of startups, given the information on startups and their founders. When new startups are being considered by a Venture Capital firm, it can simply add them and their features to the graph structure. After training, which generally takes less than a minute, we can predict the probability of success for these startups.

At the moment, the framework only considers 7 attributes of one founder for each company due to time limit, and thus there is a lot of room for improvements. The future directions to improve this framework are as follows:

- We can always include more features such as locations, the number of languages spoken, and so on.
- Multiple founders should be considered for each company, and multiple educational backgrounds and work experiences should be considered for

each founder. This can be achieved through grouping or the use of deep set [5].

- We made the assumption that the relationship between startups is only through investors and there is no weight attached to each edge. We can explore other connections such as alumni networks and geographical locations. At the same time, the connection between two startups can be strong or weak depending on these factors.
- We can experiment with more advanced machine learning models such as Graph Isomorphism Networks (GIN) [6].
- We might not have to remove nodes that are less connected to the main subgraph through methods such as expander graph propagation [7] and master nodes [8]. It means that more companies can be included in the analysis.
- etc.

In conclusion, this framework provides a starting point for future research and applications. Given limited data, our study has demonstrated promising results that suggest the potential of graph neural networks to predict a startup's success. The limitations of our study provide a guide for future research to address these challenges and to build upon our findings. We hope that our work will contribute to the advancement of VenTech and inspire new ideas and approaches to discover high-potential startups.

References

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[2] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[3] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," *arXiv preprint arXiv:2006.05205*, 2020.

[4] S. Haykin, *Neural networks: a comprehensive foundation.* Prentice Hall PTR, 1998.

[5] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," *Advances in neural information processing systems*, vol. 30, 2017.

[6] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.

[7] A. Deac, M. Lackenby, and P. Veličković, "Expander graph propagation," *arXiv preprint arXiv:2210.02997*, 2022.

[8] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*, pp. 1263–1272, PMLR, 2017.