ANALYSING NODE POLYSEMY IN NETWORKS

Gautam Choudhary
Computer Science Dept.
Purdue University
gchoudha@purdue.edu

ABSTRACT

Network embedding is a powerful technique used in graph mining that involves representing nodes in a graph as vectors. However, the traditional approach of learning a single vector for each node has been criticized for its inability to fully capture the multiple *aspects* (or facets) of a node. To address this issue, many existing studies rely on pre-computed graph clustering. Still, they are not useful as they result in a fixed distribution of aspects for each node throughout the training procedure. In this paper, I investigate a state-of-the-art multi-embedding approach ASP2VEC Park et al. (2020) under the motivation of node polysemy, wherein a node in a graph exhibits different behaviors in different contexts. Moreover, I also compare and analyze two other related baselines (NODE2VEC and DEEPWALK) which also fall under the same category of random-walk-based approaches for node embedding. The utility of multi embedding approach is shown through an empirical study with datasets from various domains. The results show the effectiveness of ASP2VEC with other baseline methods under the assumption that the graph data has inherently many latent-factors or *aspects* to capture.

1 INTRODUCTION

Network (graph) data is ubiquitous in the present scenario, from social network of Facebook friends to protein interactions in molecules to relational databases consisting of heterogeneous entities. Network Representation Learning aims to learn continuous vector representations of network data (typically nodes), which is originally in the non-euclidean space. These learned node embeddings encode the structural and semantic information of the node and its neighborhood in a low-dimensional vector space. This embedding can then be used as input to downstream tasks such as node classification , link prediction , or graph clustering .

Much research has been aimed at analyzing and extracting knowledge from networks over the past few decades. Network embedding has recently gained much attention among the various techniques used for network analysis. Many network embedding methods share the idea that the vector representation of a node should capture its local structural features, which can be used to preserve its neighborhood structure. One of the earliest and most influential methods in this field is Deepwalk Perozzi et al. (2014), which uses co-occurrence information to learn node representations. Essentially, co-occurrence based methods rely on random walks to generate node sequences that can be fed into a skip-gram model, which can then learn vector representations for each node based on the patterns of its co-occurrences with other nodes.

However, existing node embedding approaches generally focus on learning a *single* embedding for a node that may not be able to capture its holistic behavior or are not representative of its multiple *facets* or *aspects*. This is because a node in a real-world scenario may show polysemous behavior in different contexts, similar to the phenomenon of word polysemy Athiwaratkun & Wilson (2017) in Language Modeling (‘bank’ can represent a financial institution or the land alongside a river). Similarly, consider a network of users and the movies they have watched. An active user may have watched multiple genres of movies and thus can have different tastes and preferences in each genre. Thus, modeling a single user representation implies either condensing of information or, worse, information loss; hence, a multi-embedding approach is desirable to capture those diverse facets. This line of research is closely connected with disentangled representation learning, which deals with separating representation dimensions based on different factors affecting the data. Another advantage of disentangling the multiple facets into multiple embeddings is improved interpretability for downstream applications.

In this work, I aim to investigate the phenomenon of node polysemy by understanding a state-of-the-art multiple node embedding approach ASP2VEC Park et al. (2020) proposed in the literature and comparing it with other similar but single embedding approaches, namely NODE2VEC Grover & Leskovec (2016) and DEEPWALK Perozzi et al. (2014). The goal is to understand their approach, assumptions they make, their limitations, and finally, experiment them on some real-world datasets for empirical analysis. My contributions are:

- Understanding and performing a comparative analysis of various random-walk based node embedding methods: a multi-embedding approach (ASP2VEC) and two single embedding approaches (NODE2VEC and DEEPWALK).
- Adapting scattered implementations into a unified framework where the experimental setup kept same for a fair comparison across these approaches. Our code can be found at: `https://github.com/gtmdotme/node_polysemy`.
- Empirical analysis of these approaches on various datasets for a general network representation learning task (link prediction).

2 RELATED WORK

Recent advances in approaches for network embeddings have received much attention due to their effectiveness in capturing both local and global contexts of the networks. These approaches for learning network representations can be generally grouped as follows:

Single Embedding Approaches. There exists significant work in developing these methods where the goal is to compute a single universal reprsentation. Node2vec Grover & Leskovec (2016) and Deepwalk Perozzi et al. (2014) represent some popular random walk based approaches where nodes are treated as words and sequences of text are generated by performing random walks on the graph. A word embedding model, such as skip-gram, is trained to learn the corresponding node embedding. In another line of work, various Graph Neural Networks (GNNs) approaches have gained significant attention which is based on message passing paradigm and the core idea is to represent a node's embedding in terms of its own and that of its neighbors. GraphSAGE Hamilton et al. (2017), GAT Veličković et al. (2017), GCN Kipf & Welling (2016), etc. are common choices for GNN architectures. Each of these works represents the node through a single embedding.

Multi Embedding Approaches. More recently, works by Park et al. (2020); Liu et al. (2019); Epasto & Perozzi (2019), which propose to learn multiple embeddings, have shown to perform better than these single embedding approaches. In these works, each latent behavior or 'aspect' of each of the nodes is represented using a single embedding vector rather than a combined embedding. To explain in more detail, PolyDeepwalk Liu et al. (2019) is a network embedding method that first determines the aspect distribution of each node using matrix factorization-based clustering and random walk sequences are then used to sample the aspect of the target and context nodes independently based on their aspect distribution, which can result in different aspects for the target and context nodes. As the name suggests, it is the successor of DEEPWALK. Splitter Epasto & Perozzi (2019) uses local clustering on a node's ego network to create multi-aspect representations, but it trains all aspect embedding vectors to be close to the original node embedding vector without considering the aspect of the target node.

3 METHOD

For investigating node polysemy, I analyze ASP2VEC, a state-of-the-art method for generating multiple embeddings for a node in the graph. They use the terminology of *aspects* and thus the name Aspect-to-Vec.

Task: Given an input graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ denotes the set of n nodes $\{v_i\}_{i=1}^{n}$ and $\mathcal{E}$ denotes the edgeset $\{e_{i,j}\}_{i,j}^{n}$ such that $e_{i,j}$ represents a connection between nodes v_i and v_j, the aim is to learn a multi-aspect embedding matrix $\boldsymbol{Q}_i \in \mathbb{R}^{K \times d}$ for each node v_i where K is an arbitrary input to specify the number of aspects and d is the embedding dimension for each aspect. These embeddings should capture the a) network structure, b) multiple aspects associated with node v_i and c) model interactions among these aspects.

Objective: Similar to other random walk-based approaches, ASP2VEC also has a similar optimization of maximizing the log-likelihood with the inclusion of aspects as stated in Eqn 1, where $p(\delta(v_i) = s)$ denotes the probability of v_i being selected to belong to the aspect s and $\sum_{s=1}^{K} p(\delta(v_i) = s) = 1$. Also, $p(v_j|v_i, \delta(v_i) = s)$ denotes the probability of a context node v_j in the neighborhood $\mathcal{N}(v_i)$ of given target node v_i whose aspect is s. Given a target node embedding vector $\boldsymbol{P}_i \in \mathbb{R}^d$, and its currently selected aspect $\delta(v_i) \in \{1, 2, ..., K\}$, our goal is to predict its context embedding vectors for the selected aspect $\delta(v_i)$, i.e., $\{Q_j^{(\delta(v_i))}|v_j \in \mathcal{N}(v_i)\}$. For each walk $\boldsymbol{w}$, maximize

$$\mathcal{L}^{(\boldsymbol{w})} = \sum_{v_i \in \boldsymbol{w}} \sum_{v_j \in \mathcal{N}(v_i)} \sum_{s=1}^{K} p(\delta(v_i) = s) \log p(v_j|v_i, \delta(v_i) = s) \tag{1}$$

where

$$p(v_j|v_i, \delta(v_i) = s) = \frac{\exp \langle \boldsymbol{P}_i, \boldsymbol{Q}_j^{(s)} \rangle}{\sum_{j' \in \mathcal{V}} \exp \langle \boldsymbol{P}_i, \boldsymbol{Q}_{j'}^{(s)} \rangle}. \tag{2}$$

The aspect selection module, as proposed by authors, determines the aspect selection probability $p(\delta(v_i) = s)$ by assuming that it can be determined by analyzing its neighbors $\mathcal{N}(v_i)$. Formally,

$$p(\delta(v_i) = s) \equiv p(\delta(v_i) = s|\mathcal{N}(v_i)) = \frac{\exp \langle \boldsymbol{P}_i, \text{Readout}^{(s)}(\mathcal{N}(v_i)) \rangle}{\sum_{s'=1}^{K} \exp \langle \boldsymbol{P}_i, \text{Readout}^{(s')}(\mathcal{N}(v_i)) \rangle} \tag{3}$$

where the $\text{Readout}^{(s)}$ is an aggregation function (such as an average, sum, etc.) that summarizes the information in the local context of node v_i with respect to aspect s.

4 EXPERIMENTS

4.1 DATASET

In order to evaluate and compare the two kinds of approaches (single vs. multi-embedding), we choose a good mix of publicly available graph datasets based on nature (directed or undirected), scale (small or large), and connectivity (sparse or dense). They can be broadly classified into academic, social, biological protein, and word-co-occurrence networks. Some high-level qualitative descriptors of these datasets are shown in Table 1. Choosing a large set of diverse datasets helps us distinguish the approach well-suited for the domain as there is no single winner for every data domain (No Free Lunch!).

4.2 BASELINES

- DEEPWALK Perozzi et al. (2014): This is a random walk-based approach that uses a skip-gram model to learn node embeddings where a node is modeled as a word in a sequence (here, random walk). The objective to maximize is:

$$\text{For each walk } \boldsymbol{w}\text{, maximize } \mathcal{L}^{(\boldsymbol{w})} = \sum_{v_i \in \boldsymbol{w}} \sum_{v_j \in \mathcal{N}(v_i)} \log p(v_j|v_i) \tag{4}$$

  where $p(v_j|v_i, \delta(v_i) = s) = \text{Softmax}(\langle \boldsymbol{P}_i, \boldsymbol{Q}_j \rangle)$ and $\langle \boldsymbol{P}_i, \boldsymbol{Q}_j \rangle$ denotes the inner product of target and context node embedding respectively.
- NODE2VEC Grover & Leskovec (2016): This is also similar to the Deepwalk method except that instead of generating random walks independently, they are generated in a biased manner where additional hyperparameters p (return parameter) and q (in-out parameter) are used to control exploration vs staying back in the local neighborhood. The objective function is the same that in Equation 4.

Dataset	**Nodes**	**Edges**	**Nature**	**Scale**	**Connectivity**	**Domain**
Filmtrust	610	1,055	Directed	Small	Sparse	Social Networks
CiaoDVD	4,562	20,037	Directed	Small	Dense	
Wiki-vote	7,066	51,832	Directed	Medium	Dense	
BlogCatalog	10,312	166,992	Un-directed	Large	Dense	
Cora	2,485	3,290	Un-directed	Small	Sparse	Academic Networks
ca-AstroPh	17,903	98,515	Un-directed	Large	Dense	
ca-HepTh	8,638	13,555	Un-directed	Medium	Sparse	
4area	20,111	30,440	Un-directed	Large	Sparse	
PPI	3,852	19,349	Un-directed	Small	Dense	Biological Networks
Wikipedia	4,777	46,256	Un-directed	Small	Dense	Text Corpus

Table 1: Characteristics and statistics (after pre-processing) of graph datasets.

For the implementation of these baselines, Gensim Rehurek & Sojka (2011) Python Library was used along with another implementation of NODE2VEC [1]. The hyperparameters used for ASP2VEC are the best as suggested by its authors. For baselines, same set of hyperparameters were used that were common to one another. All the experiments were carefully seeded for reproducibility.

4.3 TASK: LINK PREDICTION

To investigate the efficacy of each method, I decide to use link prediction as the downstream task. Link prediction measures and learns to predict the affinity between two nodes to form an edge (link). This is an ideal way to assess these node embedding methods because, at the core, they are trying to learn the representations from these interactions happening within the network. Link prediction is also the recommended primary evaluation method for unsupervised network embedding approaches compared to node classification. I use a method similar to Epasto & Perozzi (2019); Grover & Leskovec (2016) for this task. Firstly, we divide the original graph into two sets of edges, $\mathcal{E}_{\text{train}}$ and $\mathcal{E}_{\text{test}}$, each of the same size. $\mathcal{E}_{\text{test}}$ is obtained by randomly removing edges while ensuring that the original graph remains connected. We then generate a set of negative edges, twice in size of $\mathcal{E}_{\text{test}}$, which is split into $\mathcal{E}_{\text{train-neg}}$ and $\mathcal{E}_{\text{test-neg}}$. Finally, we train a simple logistic regression classifier on $\mathcal{E}_{\text{train}} \cup \mathcal{E}_{\text{train-neg}}$, and then measure the link prediction performance on predicting the removed edge set $\mathcal{E}_{\text{test}} \cup \mathcal{E}_{\text{test-neg}}$.

Dataset	**DEEPWALK**	**NODE2VEC**	**ASP2VEC**
Filmtrust	0.7280	**0.7550**	0.7309
CiaoDVD	0.7162	**0.7549**	0.7387
Wiki-vote	0.5782	**0.6474**	0.6459
BlogCatalog	0.7376	0.7053	**0.9532**
Cora	0.8072	0.8036	**0.8825**
ca-AstroPh	**0.9770**	0.9758	0.9727
ca-HepTh	0.8917	0.8963	**0.9024**
4area	0.9348	**0.9445**	0.9423
PPI	0.7314	0.7102	**0.8784**
Wikipedia	0.5643	0.5724	**0.9074**

Table 2: Link Prediction Performance (ROC-AUC) on various datasets and methods.

[1] https://github.com/eliorc/node2vec

4.4 Results and Discussion

Table 2 reports the results for the link prediction task on various datasets. I used the original authors' implementation for Asp2vec and additionally implemented the baselines. While I was successfully able to reproduce the results for Asp2vec model but surprisingly, the performance of baseline models became at par with the multi-embedding approach in some cases. We can still make some general observations:

- From the perspective of computational time complexity, I observe that Deepwalk and Node2vec generally train faster compared to Asp2vec owing to their simpler model design. Between Deepwalk and Node2vec , the latter usually performs marginally better though the former trains faster. This tradeoff between time and performance can be an important choice for model selection.
- Between single and multi-embedding, the latter remains highly competitive in most of the datasets but also wins by a substantial margin in cases like BlogCatalog, Cora, PPI, Wikipedia, etc. It suggests that Asp2vec can learn better when the graph is small and more densely packed. Note that the learnable parameters for Asp2vec increases with the number of nodes in orders of K (no. of aspects). Hence, for larger graphs, this model trains the slowest due to a large number of model parameters to optimize over.
- In most of the datasets like biological, and text (word co-occurrence) networks, Asp2vec generally performs better than most of the baselines. This is because the nodes in these networks are inherently richer in diverse aspects than those for academic networks where a node (author or publication) is usually more focused. Perhaps this is why we observe very less variability in scores for academic datasets since there are very few aspects to be captured in those networks.

5 Conclusion

In this work, I have investigated the phenomenon of node polysemy by understanding a state-of-the-art multiple node embedding approach Asp2vec proposed in the literature and comparing it with other similar but single embedding approaches, namely Node2vec and Deepwalk. The goal is to understand their approach, assumptions they make, and their limitations, and finally, experiment with them on some real-world datasets for empirical analysis. In the future, it would be interesting to see a broader survey of other useful methods proposed in the literature for representation learning, such as those including Graph Neural Networks (GNNs). This survey could also benefit from a qualitative analysis as it will strengthen the need for a multi-embedding approach. Finally, in view of the complexity constraints of Asp2vec , the need of designing a computationally efficient approach is still a promising avenue.

Acknowledgments

The following two papers, namely Asp2vec and PolyDeepwalk , have greatly helped me understand the landscape of various node embedding methods. I want to acknowledge their work in forming my understanding of node polysemy.

References

Ben Athiwaratkun and Andrew Gordon Wilson. Multimodal word distributions. *arXiv preprint arXiv:1704.08424*, 2017.

Alessandro Epasto and Bryan Perozzi. Is a single embedding enough? learning node representations that capture multiple social contexts. In *The world wide web conference*, pp. 394–404, 2019.

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

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Ninghao Liu, Qiaoyu Tan, Yuening Li, Hongxia Yang, Jingren Zhou, and Xia Hu. Is a single vector enough? exploring node polysemy for network embedding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 932–940, 2019.

Chanyoung Park, Carl Yang, Qi Zhu, Donghyun Kim, Hwanjo Yu, and Jiawei Han. Unsupervised differentiable multi-aspect network embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1435–1445, 2020.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2):2, 2011.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.