Motif-Aware Graph Embeddings

Anonymous authors

Abstract

In this paper, we propose two motif-aware approaches for the unsupervised and semi-supervised graph embedding task. Our first model applies the most significant motif pattern on a graph as a guilding pattern for random walks. We then use a skipgram model with noise contrastive estimation to learn the graph embedding from generated random walk context. The second model employs the higher-order organization (i.e. motifs organization) of complex networks, and injects the higher-order connectivity patterns into each layer in a deep graph convolutional networks. We demonstrate the effectiveness of our motif-aware approaches on node labels classification, link prediction, and t-SNE visuallization.

1 Introduction

1.1 Complex network and machine learning

Network modeling have been an essential tool for a wide range of scientific fields [Newman, 2010; Bader *et al.*, 2003; Tang *et al.*, 2012; Milo *et al.*, 2002; Benson *et al.*, 2016]. The network science view usually reveals the underlying structure of a complex system. Based on the system's network structure, scientists can make predictions and explaination about the system's behavior. For example, in biology, the study on neuronal systems connectivity indicated that the component arrangement of a neural system is optimized for short processing paths rather than wiring lengths [Kaiser and Hilgetag, 2006]. Similarly, social networks analysis provides communities structures as well as social interaction patterns [West *et al.*, 2014; Barabási, 2014]. However, along with the information explosion, the large graph-structured data poses a great challenge for traditional network analysis methods in term of scalability and complexity. To deal with such challendges, one promising approach is to apply machine learning methods (especially deep learning) methods to network problems.

Bridging the gap between network science and machine learning is also a challenging task. Due to the irregularity in network and graph-structured data, it is desirable to have a *meaningful* and structural network representation for machine learning application. Traditionally, vector representation can be obtained via graph spectral methods. However, spectral methods are shown to be unscalable without estimation methods TODO: find theoretical citation [Perozzi *et al.*, 2014; Grover and Leskovec, 2016]. Recently, inspired by the skipgram model in natural language processing [Mikolov and Dean, 2013], Perozzi *et al.* propsed their scalable graph embedding algorithm named DeepWalk. Their results node classification proved the effectiveness of their algorithm in learning a lower dimensionality representation of a complex network. Subsequence works to DeepWalk further improved node classification accuracy by modifying graph context generation process [Tang *et al.*, 2015; Cao *et al.*, 2015; Grover and Leskovec, 2016]. On the other hand, more direct (and more effective) approaches were proposed in [Yang *et al.*, 2016; Kipf and Welling, 2016]. Instead of learning the network representation using only network structure (e.g. adjacency matrix), Yang *et al.* proposed to injects the known labeling and node feature into the representation learning process. Kipf and Welling further improved results from planetoid [Yang *et al.*, 2016] by applying graph convolution technique in their deep network model. These aforementioned approaches are similar in the sense that they all learn a latent representation of a complex network from data, then use this representation to solve a network problem using various machine learning tools.

1.2 Motifs in complex network

There are three scale of network analysis: macroscopic, mesoscopic, and microscopic. The macroscopic scale displays a network as a whole to study its robustness [Callaway *et al.*, 2000] or dynamics TODO: find citation [Barabási, 2014]. In contrast, the microscopic scale studies the pair-wise interactions between nodes in a network which is specific to the given system TODO: find citation [Newman, 2010]. On the other hand, the mesoscopic scale is an intermediate in which we consider the network is a composition of subgraphs. In many research, especially computational biology, the mesoscopic components are called *motifs*, and it is common to think of them as building blocks for a complex system [Milo *et al.*, 2002].

Definition 1.1. *Network motif* Given a graph $G = V, E$, define a subgraph $G' = V', E'$ with $V' \subseteq V$; $E' \subset E$ s.t. $i, j \in V' \forall e_{ij} \in E'$ and $|V'| \ll |V|$. Recurring subgraphs are called *network motif* when they are statistically significant.

Also refered as higher-order organization by Benson *et al.*, network motifs are believed to represent the underlying mechanism of a complex system [Alon, 2007; 2006; Mangan and Alon, 2003]. For instance, the directional bi-fan motif TODO: figure and its simplified undirectional version TODO: figure are crucial in a citation network. Beside having a statistical significance, bi-fan motif is also intuitively sensible in citation network as it represents the citation mechanism as an activity in a subgraph. The correlation of recurring subgraphs and system functionality has been studied extensively in biological systems such as transcription networks [Mangan and Alon, 2003] and brain networks [Van Den Heuvel and Pol, 2010; Honey *et al.*, 2007]. As networks motifs have been recognized as the fundamental building block of a complex systems, using them as a strucutural guidance for machine learning on graph data can yield possitive improvements.

Generally, algorithms involving network motifs have to deal with the problem of graph isomorphism. For such reason, in most analysis, only motifs of size 5 or smaller are considered. In this paper, we only consider motif of size 4 at most. This limitation is due to the large size of networks that we experimented. Although limited by the motif size, we have been able to practically show the effectiveness of the motif-aware methods. On the other hand, as mentioned in [Benson *et al.*, 2016], motif algorithms can be easily parallelized. Therefore, the extension to larger size motifs can be made possible by parallelize the motif analysis procedures. Further discussion will be provided in later sections.

2 Methods

In this section, we present the detail of our methods. Firstly, we propose the basis for the network motif selection from a network. Secondly, we present two approaches employing motif patterns to learn graph embeddings: *motifwalk* and *m-gcn*.

2.1 Network Motifs

In the previous section, we have introduced the importance of network motifs in network analysis. In this section, we present the metric for measuring network motif significance and the definition of motif laplacian.

In order to measure the importance of a network motif, we compare the given network against a null model. The null model of an empirical network is an ensemble of randomly generated networks having the same number of nodes and edges as the network. For small networks with less than 10,000 edges, we generated 100 random networks as the ensemble of the null model. On the other hand, we generated 10 random networks for the null model of larger networks. The z-score is given by:

$$z\text{-score} = \frac{N_{\mathbf{m}}(G) - N_{\mathbf{m}}(G_{\text{random}})}{\sigma_{\mathbf{m}}(G_{\text{random}})}$$

where $N_{\mathbf{m}}(G)$ is the count of motif $\mathbf{m}$ in the empirical network; $N_{\mathbf{m}}(G_{\text{random}})$ is the mean of the null model; and $\sigma_{\mathbf{m}}(G_{\text{random}})$ is the variance. The z-score's values can range from $-\inf$ to $+\inf$. In practice, the most simple motifs (figure **??**-m2,3,4) often have the highest frequencies and negative z-score. We ignored such motifs in our analysis. We select motif which has the highest positive z-score and the highest frequency as our motif of interest to construct the motif co-occurrence matrix.

The formal definition of the motif co-occurrence matrix for a motif $\mathbf{m}$ on an unweighted, directed graph G is given by:

$$M_{i,j} = \sum_{(v,\chi_{\mathcal{A}}(v))\in\mathbf{m}} \mathbf{1}(i,j \subset \chi_{\mathcal{A}}(v))$$

In here, $\mathcal{A}$ represents the anchor set; $(v, \chi_{\mathcal{A}}(v))$ represents pairs of node $v \in V_G$ and the other anchor nodes generated by $\chi_{\mathcal{A}}$. If the anchor node set $\mathcal{A}$ is empty, all motif co-occureence is counted toward the motif co-occurrence matrix M. Otherwise, only nodes in the anchor set will be counted. Figure **??** illustrates the bi-fan motif and an anchor set.

Generally, in this paper we employ the motif co-occurrence matrix as: 1. An adjacency matrix describing a motif graph; 2. An adjacency matrix from which we computes the Fourier basis for the graph convolution operation. The first approach is straight forward as we want to generate a network context where nodes occur in the motif pattern. For such reason, we treat the motif co-occurence matrix as a binary matrix describing a new network. On the other hand, the graph convolution approximation methods proposed in [Kipf and Welling, 2016; Defferrard *et al.*, 2016] only apply to symmetric binary matrices. In our model, the motif co-occurence matrix is a symmetric weighted matrix. The eigenvalue decomposition of such matrix is given by:

$$\mathcal{L}_{\mathbf{m}} = U_{\mathbf{m}} \Lambda_{\mathbf{m}} U_{\mathbf{m}}^{\top} \tag{1}$$

where $\mathcal{L}_{\mathbf{m}} = D_{\mathbf{m}} - A_{\mathbf{m}}$; $U_{\mathbf{m}}$ is the orthogonal basis (also called the Fourier basis in graph convolutional context); and $\Lambda_{\mathbf{m}} = diag(\lambda_{\mathbf{m}})$.

The convolution on a graph G of a function of the graph Laplacian g_θ (also called a filter or a kernel) and a signal x is defined as:

$$g_\theta * x = U g_\theta U^{\top} x,$$

where $L = U\Lambda U^{\top}$, U is the Fourier basis and Λ is called the frequencies of the graph. Graph convolution has been shown effective in processing graph-structured data, and also argued to be the generalization of convolutional networks [Shuman *et al.*, 2013; Defferrard *et al.*, 2016; Kipf and Welling, 2016]. In practice, given a graph where each node has a feature vector, we can treat the feature vector of the graph as signals. The output y of these "signals" filtered by g_θ on the graph is given by the graph convolution and deconvolution operations:

$$y = g_\theta(U\Lambda U^{\top})x = U(g_\theta(\Lambda)U^{\top})x \tag{2}$$

Computing equation 2 is computationally expensive due to the matrix multiplication and eigenvector decomposition operations. Therefore, fast estimation methods such as Chebyshev polynomial was suggested in [Hammond *et al.*, 2011].

TODO: Algorithms and estimation techniques. Discuss about the change in maximum eigenvalue due to the weighted motif laplacian.

2.2 Biased Random Walk

Previous skipgram-based graph embedding models employ random walks for graph context generation. To improve

Algorithm 1: Motif co-occurrence matrix generation

```
Data: Graph G = (V, E)
Input: binary, m, A
Output: context
begin
    context ← [ ];
    V ← G.nodes();
    nodes ← Shuffle(V);
    for node ∈ nodes do
        walks ← [ ];
        for i=0; i¡nwalk; ++i do
            walks += RandomWalk(graph=G,start=node,
              len=length)
        for i=0; i¡nwalk; ++i do
            walks +=
              RandomWalk(graph=G_m start=node,
              len=length)
        context += walks
    return context
```

Algorithm 2: Motif-aware graph context generation

```
Data: Graph G = (V, E), Motif Graph G_m = (V, E)
Input: length, nwalk, nmwalk
Output: context
begin
    context ← [ ];
    V ← G.nodes();
    nodes ← Shuffle(V);
    for node ∈ nodes do
        walks ← [ ];
        for i=0; i¡nwalk; ++i do
            walks += RandomWalk(graph=G,start=node,
              len=length)
        for i=0; i¡nwalk; ++i do
            walks +=
              RandomWalk(graph=G_m start=node,
              len=length)
        context += walks
    return context
```

the embedding results, structure-aware context generation methods were proposed in [Tang *et al.*, 2015; Grover and Leskovec, 2016]. However, the limitation of *LINE* lies at the fact that it only consider the second-order proximity (bi-fan motif), *node2vec* requires the costly cross-validation search for its hyperparameters p and q. To solve the above mentioned problems, we propose a biased random walk algorithm for graph context generation which can be considered the generalization of *LINE* and *deepwalk*. Since our algorithm decides the walk pattern supported by the most significant network motif before performing context generation, we achieve the simplicity of *deepwalk* while having the structure-aware context as of *LINE* and *node2vec*.

Our *motifwalk* algorithm has two steps: motif adjacency matrix construction and context generation. Firstly, we construct a binary motif co-occurence matrix from the given network. We select the motif pattern as described in the previous section. Since the constructed matrix accounts the co-occurrence of network node pairs in a motif, it is a symetric matrix. Secondly, after having a second adjacency matrix describing the motif structure, we run random walks on this new network for context generation. The obtained context is used jointly with random walks context generated with the original network to train an embedding skipgram model. Algorithm 1 and algorithm 2 describle the *motifwalk* algorithm.

2.3 Motif Convolutional Architecture

In this section we propose our motif convolutional deep neural network architecture for semi-supervised graph labeling tasks. Graph convolution is a signal processing technique in which a network of signals reside on nodes (e.g. sensor network) is processed in the graph spectral domain defined on the graph structure. Based on the further linear approximation proposed by Kipf and Welling, we propose our two layers motif convolutional network model as:

$$\begin{aligned} Z_{\text{forward}} &= f(X, A, M) \\ &= \text{softmax}(\hat{M}\text{ReLU}(\hat{A}XW^{(0)})W^{(1)}), \end{aligned} \tag{3}$$

where A and M is a binary adjacency matrix and motif co-occurrence matrix respectively; $\hat{A}$ and $\hat{M}$ are constructed by the *renormalization trick* as suggested in [Kipf and Welling, 2016]; X contains the feature vectors for each graph node; $W^{(0)}$ and $W^{(1)}$ are learnable variables. With the backpropagation learning algorithm, the weight of layer k is updated as follow:

$$\frac{\partial E}{\partial W_{i,j}^{(k)}} = \sum_{s=1}^{S} [x]^\top \frac{\partial E}{\partial y_{k,j}} \tag{4}$$

3 Experiments

3.1 Datasets and observations

3.2 Motif significance

4 Results

4.1 Unsupervised

Traditional task on blogcatalog and others. Link prediction. t-SNE.

4.2 Semi-supervised

Task on featured networks.

5 Related work

5.1 Spectral approaches

5.2 Skipgram-based approaches

5.3 Deep neural network approaches

6 Discussion

Our paper's contributions are proposing an extension to the graph convolutional architecture; proposing the uses and demonstrate the importance of motifs in real worldnetworks.

Limitation:

Acknowledgments

I would like to thank.

References

[Alon, 2006] Uri Alon. *An introduction to systems biology: design principles of biological circuits*. CRC press, 2006.

[Alon, 2007] Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

[Bader *et al.*, 2003] Gary D Bader, Doron Betel, and Christopher WV Hogue. Bind: the biomolecular interaction network database. *Nucleic acids research*, 31(1):248–250, 2003.

[Barabási, 2014] Albert-László Barabási. Network science book. *Network Science*, 2014.

[Benson *et al.*, 2016] Austin R Benson, David F Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

[Callaway *et al.*, 2000] Duncan S Callaway, Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Network robustness and fragility: Percolation on random graphs. *Physical review letters*, 85(25):5468, 2000.

[Cao *et al.*, 2015] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891–900. ACM, 2015.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3837–3845, 2016.

[Grover and Leskovec, 2016] 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*, 2016.

[Hammond *et al.*, 2011] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[Honey *et al.*, 2007] Christopher J Honey, Rolf Kötter, Michael Breakspear, and Olaf Sporns. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proceedings of the National Academy of Sciences*, 104(24):10240–10245, 2007.

[Kaiser and Hilgetag, 2006] Marcus Kaiser and Claus C Hilgetag. Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems. *PLoS Comput Biol*, 2(7):e95, 2006.

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

[Mangan and Alon, 2003] Shmoolik Mangan and Uri Alon. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21):11980–11985, 2003.

[Mikolov and Dean, 2013] T Mikolov and J Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.

[Milo *et al.*, 2002] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[Newman, 2010] Mark Newman. *Networks: an introduction*. Oxford university press, 2010.

[Perozzi *et al.*, 2014] 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*, pages 701–710. ACM, 2014.

[Shuman *et al.*, 2013] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

[Tang *et al.*, 2012] Lei Tang, Xufei Wang, and Huan Liu. Scalable learning of collective behavior. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):1080–1091, 2012.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, 2015.

[Van Den Heuvel and Pol, 2010] Martijn P Van Den Heuvel and Hilleke E Hulshoff Pol. Exploring the brain network: a review on resting-state fmri functional connectivity. *European Neuropsychopharmacology*, 20(8):519–534, 2010.

[West *et al.*, 2014] Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. Exploiting social network structure for person-to-person sentiment analysis. *arXiv preprint arXiv:1409.2450*, 2014.

[Yang *et al.*, 2016] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning*. ICML, 2016.