Motif-Aware Graph Embeddings

Hoang Nguyen and Tsuyoshi Murata
School of Computing, Computer Science Department
Tokyo Institute of Technology

Abstract

In this paper, we propose our motif-aware approaches to the unsupervised network embedding and semi-supervised network labeling task. Our first algorithm is an unsupervised network embedding algorithm which uses the most statistically significant network motif as the guiding pattern for random walks to generate network context. We then use a Skipgram neural network to learn the latent network node representations from the generated context via Noise Contrastive Estimation. The second algorithm employs the Graph Convolution Network model on motif Laplacian matrices to inject the higher-order network structure into the neural network. Both of our algorithms utilize the higher-order organization (i.e. motifs organization) of complex networks. We demonstrate the effectiveness of our algorithms in comparison with other state-of-the-art network embedding algorithms.

1 Introduction

1.1 Complex network and machine learning

Network modelings have been an essential tool for a wide range of scientific fields Newman [2010]; Bader et al. [2003]; Milo et al. [2002]; Benson et al. [2016]. Based on the system’s network structure, scientists can make predictions and explanation about the system’s behavior. For example, in biology, the study of neuronal system connectivity indicated that the arrangement of neurons is optimized for short processing paths rather than wiring lengths Kaiser and Hilgetag [2006]. Similarly, social network analysis provides community structures well as social interaction patterns West et al. [2014]; Barabási [2014]. However, along with the information explosion, analyzing large network-structured datasets poses a great challenge for traditional network analysis methods in term of scalability and complexity. To deal with such challenge, one promising approach is to apply machine learning methods (especially deep learning) to network problems.

1.2 Motifs in complex networks

There are three scales of network analysis: macroscopic, mesoscopic, and microscopic. In the macroscopic scale, we consider a network as a whole to study its macro-properties such as robustness Callaway et al. [2000], or dynamics Barabási [2014]. In contrast, the microscopic scale studies the pair-wise interactions between nodes in a network which is specific to a given system Newman [2010]. In between macroscopic and microscopic, the mesoscopic scale considers the network is a composition of subgraphs. In many fields of research, especially computational biology, the mesoscopic components are called motifs, and it is common to think of them as building blocks of 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 $V' \subseteq V$; $E' \subseteq E$ s.t. $i, j \in V'$, $e_{ij} \in E'$ and $|V'| \ll |V|$. Recurring subgraph is called network motif when they are statistically significant.
Also referred as the higher-order organization by Benson et al., network motifs are believed to represent the underlying mechanism of a complex system Alon [2007]; Mangan and Alon [2003]. For instance, the directional bi-fan motif (Figure 3: m4-1) and its simplified unidirectional version (m4u-3) are crucial in a citation network. This bi-fan motif is also intuitively sensible in citation network as it represents the citation mechanism. The correlation of recurring subgraphs and system functionality has been studied extensively in biological systems such as transcription networks Mangan and Alon [2007]; Honey et al. [2007]. As networks motifs have been recognized as the fundamental building block of a complex network, using them as a structural guidance for machine learning on network data is expected to yield positive improvements.

Our main idea in this paper is to construct the motif cooccurrence matrix from a given network, and use it as: 1. An adjacency matrix describing a motif network for random walks; 2. A mean to compute motif Laplacian and Fourier basis for the graph convolution operation. Section 2 describes the related work on network motif conductance and network embedding. We give detail of our algorithms in section 3. The experimental setup and results are given in section 4 and 5 respectively. We discuss the relationship between our two proposed algorithms and their limitations in section 6.

2 Related work

In this section, we introduce the recent developments regarding unsupervised network embeddings, semi-supervised node labeling, and motif analysis.

2.1 Unsupervised Network Embedding

Traditionally, network embedding can be obtained via graph factorization methods. However, matrix factorization methods such as Spectral Clustering or Non-negative Matrix Factorization are shown to be unsuitable due to the complexity of the algorithms Perozzi et al. [2014]; Belkin and Niyogi [2001]. Recently, several feasible network embedding algorithms have been proposed such as Deepwalk Perozzi et al. [2014] or node2vec Grover and Leskovec [2016]. These network embeddings algorithms learn high-quality node representation while having low time complexity compared to traditional methods. In the context of graph embedding, we justify the embedding quality by how well a common machine learning model performs on the learned embeddings.

Based on the Skipgram model Mikolov and Dean [2013] in natural language processing, Perozzi et al. proposed their scalable graph embedding algorithm named Deepwalk. Their results on node classification proved the effectiveness of Deepwalk in learning a lower dimensionality representation of a complex network. Subsequent work to Deepwalk further improved node classification accuracy by modifying graph context generation process Tang et al. [2015]; Grover and Leskovec [2016]. Generally, these algorithms optimize the following objective:

\[
\mathcal{O} = \arg\max_{\mathbf{W}_{\text{emb}}} (\ell_{\text{random walk}}),
\]

where \( \ell_{\text{random walk}} \) represents the log-likelihood of the network context generated by random walks on the given network. We have the pairwise potential between a context vertex \( v_c \) and a target vertex \( v_t \) as follow:

\[
\phi_{v_t,v_c} = \exp (\langle \omega_v, \omega_{v_t} \rangle)
\]

\[
\ell_{\text{random walk}} = \sum_{v_t,v_c} \log \left( \frac{\phi_{v_t,v_c}}{\sum_{k \in V} \phi_{v_k,v_t}} \right)
\]

In here, \( \langle \cdot, \cdot \rangle \) denotes the inner product; \( \omega_v \) denotes the real vector representation of node \( v \); and \( \mathbf{W}_{\text{emb}} \) is the output matrix containing all real vector representations. Although the log-likelihood given by equation (1) is tractable, computing the normalization factor still remains an intractable task. Therefore, several approximation methods such as hierarchical softmax Morin and Bengio [2005]; Perozzi et al. [2014] or noise contrastive estimation Gutmann and Hyvärinen [2010]; Grover and Leskovec [2016] were used to further boost the computational speed of the algorithms of this category. The obtained embedding matrix will be used as a feature matrix for various machine learning algorithms such as node classification (Multi-class Linear Regression model) or link prediction.

2.2 Semi-supervised Network Labeling

Planetoid, proposed by Yang et al. [2016], works slightly different to other Skipgram-based models. Instead of generating graph context only from the network structure, Planetoid also samples nodes based on training labels. Furthermore, Planetoid injects the network node’s feature vectors for better embedding and node labeling results. Planetoid can be considered an improvement of SemiEmb, proposed in Weston et al.
Another semi-supervised learning model similar to Planetoid is Graph Convolutional Networks (GCN) Kipf and Welling [2016]. GCN uses the graph convolutional operation as a transformation for feature vectors on a network. By stacking these convolutional operation into a neural network, the authors of GCN has been able to achieve remarkable accuracy in node classification and link prediction results compared with the previous unsupervised and semi-supervised algorithms. Moreover, the running time for GCN was shorter compared with other algorithms such as Deepwalk or Planetoid. In the following paragraphs, we present the details for GCN.

The convolution on a graph $G$ of a function of the graph Laplacian $g_\theta$ (also a filter or a kernel) and a signal $x$ is defined as:

$$g_\theta * x = g_\theta(L)x,$$

where the normalized Laplacian $L = U \Lambda U^T$; $U$ is the Fourier basis and $\Lambda$ is the frequencies of the graph. Graph convolution has been shown effective in processing graph-structured data, and also argued to be the generalization of convolution. Similar to network conductance, motif conductance is a score for a cut $(S, \bar{S})$ targeting a motif $m$:

$$\phi_m(S) = \frac{\text{cut}_m(S, \bar{S})}{\min[\text{vol}_m(S), \text{vol}_m(\bar{S})]},$$

where $S$ is a node set in a network $G$; $\bar{S}$ is the complement of $S$; $\text{vol}_m(S)$ is the number of motif instances that resides in $S$. Intuitively, minimizing the motif conductance is equivalent to minimizing the number of motifs $m$ split by the cut. A motif is split when there is at least one anchor node in $S$ and at least one in $\bar{S}$. Benson et al. then perform motif analysis and graph clustering based on this definition of motif conductance. Their result further confirms the structural role of motifs in a complex network. It also hinted that there is a strong motif structure within a community, which we can use as prior knowledge for solving problems on networks.

**Definition 2.1. Motif co-occurrence matrix:** Given a graph $G = (V, E)$, in which $v \in V$. The motif co-occurrence matrix of a motif $m$ is given by:

$$M_m = \sum_{(v, \chi_A(v)) \in m} 1(i, j \in \chi_A(v))$$

In here, $A$ represents the anchor set; $(v, \chi_A(v))$ represents pairs of node $v \in V_G$ and the other anchor nodes generated by $\chi_A$. If the anchor node set $A$ is empty, all motif co-occurrence is counted toward the motif co-occurrence matrix $M$. Otherwise, only nodes in the anchor set will be counted. Figure 3 illustrates the bi-fan motif and its anchor set. Algorithm 1 provides the detail for constructing a motif co-occurrence matrix.

### 3 Our 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-gen.

3.1 Motif Analysis and Motif Convolution

In the previous section, we have explained 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 generated 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 10,000 edges, we generated 10 random networks as the ensemble of random 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. The z-score is given by:

$$z\text{-score} = \frac{N_m(G) - N_m(G_{\text{random}})}{\sigma_m(G_{\text{random}})}$$

where $N_m(G)$ is the count of motif $m$ in the empirical network; $N_m(G_{\text{random}})$ is the mean of the null model; and $\sigma_m(G_{\text{random}})$ is the variance. The z-score’s values can range from $-\infty$ to $+\infty$. In practice, the most simple motifs (Figure 2 m3u-2, m3u-8, m3u9) often have the highest frequencies and negative z-score. We ignored such motifs in our analysis. We select the motif which has the highest positive z-score because these motifs highlight the difference between the empirical network and random networks.

Convolution operations on a network can be viewed as a method to incorporate the nodes’ information (e.g. feature vectors) and the network structure. The Fourier basis and network frequencies of a motif co-occurrence matrix is obtained through the eigenvalue decomposition of the motif Laplacian matrix $L_m$:

$$L_m = U_m \Lambda_m U_m^\top$$

In here, $\Lambda_m = \text{diag}(\lambda_m)$ is the frequencies of the motif network; $L_m$ is the normalized motif Laplacian given by:

$$L_m = D_m^{-1/2} L_m D_m^{-1/2}$$

$$= I - D_m^{-1/2} M^m D_m^{-1/2}$$

(7)

where $L_m = D_m - M^m$, $M^m$ is the motif co-occurrence matrix; and $D_m = \text{diag}(\sum_i M_{i,i})$.

$U_m$ from equation (7) is the Fourier basis of the network motif structure which is used for motif convolution. Given signal $x \in \mathbb{R}^n$ on a network, the motif Fourier transform is defined as $\hat{x} = U_m^\top x$, and its inverse as $x = U_m \hat{x}$. It follows that the output of signal $x$ filtered by a function of graph Laplacian $g_\theta(L)$ parameterized by a set of coefficient $\theta$ is given by:

$$y = g_\theta \ast x = g_\theta(U_m \Lambda_m U_m^\top) x$$

$$= U_m(g_\theta(\Lambda_m) U_m^\top)x$$

(8)

Due to the complexity of eigenvector decomposition and matrix multiplication, we use the linear formulation suggested in Kipf and Welling [2016] to estimate the costly convolution operation:

$$y = g_\theta \ast x \approx \sum_{k=0}^K \theta_k T_k(\hat{L})x$$

$$\approx \theta_0 x + \theta_1 (L - I) = \theta_0' x - \theta_1' D_m^{-1/2} M^m D_m^{-1/2}$$

(9)

3.2 Biased Random Walk

Previous Skipgram-based graph embedding models employ random walks for graph context generation. To improve 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), and node2vec requires the costly cross-validation grid search for its hyper-parameters $p$ 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 walking pattern supported by the most significant network motif before performing context generation, our method has 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-occurrence 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 symmetric matrix. Secondly, after having the motif adjacency matrix, we run random walks on this new network induced by the adjacency matrix for motif context generation. The obtained motif context is used jointly with random walks context generated with the original network to train an embedding Skipgram model. Algorithm 1 and algorithm 2 describe the motifwalk algorithm.

Similar to other Skipgram-based models, motifwalk is an unsupervised algorithm which learns graph embedding through an optimization process. Since there is two network contexts generated in our algorithm, the objective function is given by:

$$O = \arg \max \_{W^{\text{emb}}} (\gamma \ell_{\text{random walk}} + (1 - \gamma) \ell_{\text{motif walk}})$$  \hspace{1cm} (10)

where $\ell_{\text{random walk}}$ and $\ell_{\text{motif walk}}$ are the log-likelihoods of the network contexts generated by random walk and motif walk respectively; $\gamma$ is a hyper-parameter controlling the ratio between random walk. $\gamma$ is selected empirically based on the rate between edges in the motif network and edges in the given network. The likelihood of a target vertex $v_t$ in the context of vertex $v_c$ is given in equation (2). We use negative sampling with noise contrastive estimation loss as suggested in Mikolov and Dean [2013] for normalization factor estimation. The output of motifwalk is a set of real vectors representing each node in the given network. These vectors encode the underlying structural relationship between network nodes and can be used as feature vectors for link prediction and node classification.

### 3.3 Motif Convolutional Architecture

In the previous section, we have defined the graph convolution operation on motif co-occurrence matrix. We use the motif convolution as the second layer in our motif convolutional network (m-gcn). Based on the linear approximation proposed by Kipf and Welling, we define the forward computation of our model as:

$$Z_{\text{forward}} = f(X, A, M)$$

$$= \text{softmax}(\hat{M}\text{ReLU}(\hat{A}XW^{(0)})W^{(1)}),$$  \hspace{1cm} (11)

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. We use Adam Kingma and Ba [2014] with back-propagation to optimize the cross-entropy loss $E$:

$$E = - \sum_{i \in Y} \sum_{j=1}^{F} Y_{ij} \ln Z_{ij}$$  \hspace{1cm} (12)

This formulation is the same as equation (5). The weight $W$ is updated by the following rule:

$$\frac{\partial E}{\partial W_{i,j}^{(k)}} = \sum_{s=1}^{S} [x]^{\top} \frac{\partial E}{\partial y_{k,j}},$$  \hspace{1cm} (13)

where $W^{(k)}$ is the weight matrix of layer $k$; $y_{k,j}$ is the $j$th output feature map of the sample $s$.

### 4 Experiments

To compare motifwalk performance with other unsupervised graph embedding models, we have chosen the node labeling task on various type of networks (with ground-truth node labels). We compare the performance of motifwalk to Spectral Clustering, Deepwalk Perozzi et al. [2014], node2vec Grover and Leskovec [2016]. Table 1 gives the networks’ statistics. The second model $m$-gcn is a semi-supervised model for node labeling task. We present the comparison of our model with planetoid and gcn under the same experimental setting in Kipf and Welling [2016]. Table 2 shows the details of each dataset.

Blogcatalog3 Tang and Liu [2009]: This network is a blogger social network. Each node in the network represents an user. The node labels represent a blogger’s interests, each node has at least one label. Since Blogcatalog is a social network, the results obtained from motif analysis agrees with the term “friends of a friend are friends”.

---

1 Although the motif z-score for blogcatalog low, we select the triangle as a mean of applying a common knowledge into a friendship network. Further study for this particular z-score case will be conducted in the future.
5 Results

Motifwalk algorithm is performed with following parameters: Number of walks per node (\textit{nwalk}): 8; Walk length (\textit{length}): 80; Skip-gram window size: 10; Samples per target: 4; Number of motif walks per node (\textit{nmwalk}): 2. The learned embeddings are used to train a Multi-class Linear Regression model with train-test split ratio of 0.5 (for all dataset in table 1). We report the f1-macro score for comparison.

\textbf{m-GCN} model consists of two convolutional layers as described by equation (11). We train the model using Adam Kingma and Ba [2014] optimizer and dropout regularization for each layer. The accuracy scores for single label classification are given in Table 5.

6 Discussion

Our results in Table 5 have shown the effectiveness of using network motifs in learning network embeddings and classifying network nodes. The connection between two of our approaches is that both of them utilizes the network co-occurrence matrix. Motifwalk performs random walks on the induced motif network to generate a network context that is biased toward the targeted motif. This graph context generation algorithm can be seen as the generalization of Deepwalk and LINE. On the other hand, m-gcn uses the motif network as a Fourier basis for graph convolution. As a result, both approaches yields improvements compared with state-of-the-art algorithms on benchmark datasets. Furthermore, based on the result here, we believe the m-gcn model can also be used as mean to measure a motif’s significant similar to motif conductance metric proposed by Benson et al.

The weakness of our algorithms lies at building the motif co-occurrence matrix. Algorithms involving network motifs have high time complexity due to the problem of graph isomorphism Tran et al. [2014]. For such reason, in most large graph analysis, only motifs of size 5 or smaller are considered. In our experiments, we only consider motif of size 4 at most. This limitation is due to the large size of networks that we experimented. Although the analysis is limited by the motif size, we have been able to empirically show the effectiveness of the motif-aware methods. Furthermore, 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 parallelizing the motif analysis procedures.

References

Gary D Bader, Doron Betel, and Christopher WV Hogue. 
Austin R Benson, David F Gleich, and Jure Leskovec.


