Spatio-Temporal Attentive RNN for Node Classification in Temporal Attributed Graphs

Dongkuan Xu1*, Wei Cheng2*, Dongsheng Luo1, Xiao Liu1 and Xiang Zhang1*

1The Pennsylvania State University
2NEC Laboratories America, Inc.
{dux19, dul262, xxl213, xzz89}@psu.edu, weicheng@nec-labs.com

Abstract

Node classification in graph-structured data aims to classify the nodes where labels are only available for a subset of nodes. This problem has attracted considerable research efforts in recent years. In real-world applications, both graph topology and node attributes evolve over time. Existing techniques, however, mainly focus on static graphs and lack the capability to simultaneously learn both temporal and spatial/structural features. Node classification in temporal attributed graphs is challenging for two major aspects. First, effectively modeling the spatio-temporal contextual information is hard. Second, as temporal and spatial dimensions are entangled, to learn the feature representation of one target node, it’s desirable and challenging to differentiate the relative importance of different factors, such as different neighbors and time periods. In this paper, we propose STAR, a spatio-temporal attentive recurrent network model, to deal with the above challenges. STAR extracts the vector representation of neighborhood by sampling and aggregating local neighbor nodes. It further feeds both the neighborhood representation and node attributes into a gated recurrent unit network to jointly learn the spatio-temporal contextual information. On top of that, we take advantage of the dual attention mechanism to perform a thorough analysis on the model interpretability. Extensive experiments on real datasets demonstrate the effectiveness of the STAR model.

1 Introduction

A large amount of graphs have been generated because of the rapid growth of information, such as co-author graphs [Ni et al., 2018], brain graphs [Wang et al., 2017a] and post graphs [Hamilton et al., 2017]. Node classification approaches aim to classify the nodes into different categories according to graph topology and node attributes. There have been a lot of efforts on node classification [Perozzi et al., 2014; Grover and Leskovec, 2016; Kipf and Welling, 2017; Veličković et al., 2018]. node2vec [Grover and Leskovec, 2016] adopts a biased random walk to learn richer node representations and classifies nodes via a logistic regression model. GCN [Kipf and Welling, 2017] introduces a convolutional architecture directly applied on graphs and node labels are predicted by a softmax function. An attention-based network that performs node classification in graphs is proposed in [Veličković et al., 2018].

In many real-world applications, graphs are often dynamic and evolve rapidly over time. Hence, basic graph concepts are further extended to temporal graphs. Some recent progresses have been made on the node classification in temporal graphs [Du et al., 2018; Goyal et al., 2018b; Zhou et al., 2018; Zhu et al., 2018]. For example, DynamicTriad [Zhou et al., 2018] extracts the evolution patterns of temporal graphs and uses a logistic regression model to classify nodes. DNE [Du et al., 2018] extends the skip-gram based graph embedding methods into dynamic settings. The work in the literature is mostly designed for the case where graph topology evolves over time. However, a vast majority of real-world networks are coupled with a rich set of vertex attributes, which also evolve over time. As temporal and spatial dimensions are entangled, it’s desirable to design an effective approach that can jointly model the spatio-temporal contextual information for both graph topology and node attributes, at the same time provide model interpretability for the classification results.

In this work, we focus on the problem of node classification in temporal attributed graphs. Given a sequence of attributed graphs for the same set of nodes, where each node has a unique class label over a period of time, the task is to predict the labels of unlabeled nodes by learning from the labeled ones. Both graph topology and node attributes change over time. The problem is challenging for two major reasons. First, effectively modeling the spatio-temporal contextual information is hard. For the temporal attributed graphs, the evolution lies in both graph topology and node attributes. Along with the changes, the temporal and spatial dimensions are entangled. How to jointly learn the node representations on both spatial and temporal aspects simultaneously is challenging. Second, differentiating the relative importance of different factors that influence node representations for effective classification is also challenging. Different neighbors at different time periods will influence node representations diversely. As the changes in temporal and spatial dimensions...
jointly determine the node labels, it is desirable to recognize which neighbors in which time periods exert more influence on the representation learning of the target node.

To address the above challenges, we propose a spatio-temporal attentive recurrent neural network model (STAR). STAR aims to learn node representations for classification by jointly considering both the temporal and spatial patterns of the node. Specifically, STAR embeds the sampling and aggregation of local neighbor nodes into a gated recurrent unit (GRU) network [Cho et al., 2014] to jointly learn the spatio-temporal contextual information. By feeding sequential node attributes at different time periods into the GRU network, the temporal features of attributes evolution can be effectively extracted. The process is entangled with the graph topology learning, in which the neighborhood representations are sampled and aggregated to learn the spatial information of the target node. Moreover, a dual attention mechanism on both temporal and spatial aspects is developed to selectively aggregate different time periods and different neighbors, which helps to perform a thorough analysis on the model interpretability. In particular, based on the temporal attention, we can detect the time periods that are more important for the node representations for classification, while the spatial attention can help us put more emphasis on the important neighbors. To summarize, the major contributions are as follows.

- We propose STAR. STAR is the first model that can provide the model interpretability for the node classification in temporal attributed graphs.
- We design a spatio-temporal GRU network that can jointly model the temporal evolution of both node attributes and graph topology.
- We perform extensive experiments on real datasets to evaluate the effectiveness of the proposed model. The results demonstrate the effectiveness of STAR.

2 The Problem

A temporal attributed graph is a collection of snapshots of an attributed graph at different time steps, denoted by $G = (G_1, G_2, \ldots, G_T)$. $G_t = (V, A_t, X_t)$ is the graph at time step $t$. The set of nodes $V$ is fixed for all time steps. Each node has its consistent label across $T$ time steps. $A_t \in \mathbb{R}^{N \times N}$ is the adjacency matrix and $X_t \in \mathbb{R}^{N \times d}$ is the node attribute matrix. Both $A_t$ and $X_t$ are different at different time steps. Given $G$ and the labels of a subset of nodes $V_L$, the goal of node classification in temporal attributed graphs is to classify the nodes in subset $V_U$ whose labels are unknown, where $V = V_L \cup V_U$.

3 Spatio-Temporal Attentive RNN

The framework of STAR is shown in Figure 1. It includes a spatio-temporal GRU and a dual attention module. First, we introduce the basic GRU. Then, we introduce the spatio-temporal GRU and discuss the dual attention mechanism.

3.1 Gated Recurrent Unit

The GRU network is an effective approach to capture the temporal patterns of sequential data. Given a sequence of input data $x_1, \ldots, x_T \in \mathbb{R}^d$, a state vector $h_t \in \mathbb{R}^{d_h}$ is calculated for each input data by applying the following equations iteratively.

$$
\begin{align*}
    z_t &= \sigma(W_z[x_t * h_{t-1}] + b_z), \\
    r_t &= \sigma(W_r[x_t * h_{t-1}] + b_r), \\
    \tilde{h}_t &= \tanh(W_h[x_t * (r_t * h_{t-1})] + b_h), \\
    h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t,
\end{align*}
$$

where $W_z, W_r, W_h \in \mathbb{R}^{d \times (d + d_h)}$ and $b_z, b_r, b_h \in \mathbb{R}^{d_h}$ are parameters. $z_t, r_t \in \mathbb{R}^{d_h}$ are called update, reset gates separately and their values are in the range of $[0, 1]$. $b_0 = 0$. $\odot$ denotes concatenation operator. $\odot$ denotes element-wise multiplication. The concatenation of all the state vectors $[h_1 \odot \cdots \odot h_T]$ or the last state vector $h_T$ is usually used as the representation for the whole sequence. GRU can be utilized to classify the nodes in temporal attributed graphs. The intuitive way is to first feed the node attributes at different time steps into GRU to generate the node representations. Then apply a classifier to the learned representations. However, the basic approach ignores the neighborhood information of the node. Next we will introduce how to jointly integrate the neighborhood information so that the graph structural information can also be encoded for node classification.

3.2 Vector Representation of The Neighborhood

We extract a neighborhood vector for each node at each time step to represent its neighborhood information. The key idea is to aggregate the neighbors’ representations. We will consider $K$-hop neighbors.

First, we prepare the $K$-hop neighbors. Given a set of nodes $B$ to be classified, we sample the immediate neighbors of the nodes in $B$. These sampled neighbors and the nodes in $B$ together form a new set $B^0$. We do the same for $B^1$ and get another new set $B^2$. As a result, we can get a sequence of sets denoted by $B^0, \ldots, B^K$, where $B^0 = B$. The neighbors are sampled by sampling function $\mathcal{N}()$. The process is described in Algorithm 1.

Then, based on the sequence of node sets, we generate the neighborhood vectors for all the nodes in $B$. It is described in Algorithm 2, where $g_{k,v}^l$ is the representation of node $v$ after aggregating its $k$-th hop neighbors at time step $t$. The key idea is to apply aggregator $AGG_k(\cdot)$ to aggregate the neighbors’ representations (Line 5) and concatenate the aggregation with...
the node’s representation (Line 6). The new representation is
generated by a nonlinear transformation. $W_{\text{trans}} \in \mathbb{R}^{d_x \times 2d_y}$
is the transformation matrix that we need to learn. We will
introduce $AGG(\cdot)$ in Section 3.4.

3.3 Spatio-Temporal GRU

To integrate the neighborhood vector, a spatio-temporal GRU
(ST-GRU) is proposed. The intuition behind ST-GRU is that we
consider the neighborhood information besides the node
attributes and the previous state vector when generating the
new proposal to update the state vector (Eq. (8)). ST-GRU has two
advantages. The state vectors explicitly contain both the
temporal information of node attributes and the spatial
information encoded in the neighborhoods. And it is practical
to further implement a dual attention mechanism on both
temporal and spatial aspects.

ST-GRU is described in Eqs. (5)-(9). Given a sequence of node attributes $x_1, ..., x_T \in \mathbb{R}^d$ and a sequence of neighbor-
hood vectors $e_1, ..., e_T \in \mathbb{R}^{d_e}$, a state vector $h'_t \in \mathbb{R}^{d_h}$ is calculated for each time step by applying the following equa-
tions iteratively, where $h_0 = 0$.

$$z_t' = \sigma(W_z^t [x_t \oplus h_{t-1} \oplus e_t] + b_z^t),$$  (5)
$$r_t = \sigma(W_r^t [x_t \oplus h_{t-1} \oplus e_t] + b_r^t),$$  (6)
$$s_t' = \sigma(W_s^t [x_t \oplus h_{t-1} \oplus e_t] + b_s^t),$$  (7)
$$h_t' = \tanh(W_{\text{h}}^t [x_t \oplus (r_t \odot h_{t-1}) \oplus (s_t \odot e_t)] + b_h'),$$  (8)

where $b_z^t, b_r^t, b_s^t, b_h' \in \mathbb{R}^{d_h}$, $W_z^t \in \mathbb{R}^{d_x \times (d + d_e + d_y)}$, $W_r^t, W_s^t, W_{\text{h}} \in \mathbb{R}^{d_h \times (d + d_x + d_y)}$ are parameters. $z_t'$, $r_t \odot h_{t-1}$, $s_t \odot e_t$ are called update, reset, neighborhood gates respectively.

Eqs. (5)-(7) describe the calculation process of the three gates. The gates control the information when generating
the state vector. The intuition of considering the node attributes,
the previous state vector and the current neighborhood vector
is that the current state vector is influenced by the current
information of the node, the previous state vector and the cur-
rent neighborhood information.

Eq. (8) describes the calculation process of a new proposal.
The values in all gates are in the range of $[0, 1]$. $r_t \odot h_{t-1}$ indicates
how much information to keep from the previous state

$$w_k \in \mathbb{R}^{d_x}$ and $v_k \in \mathbb{R}^{d_y}$ are parameters.
Table 1: Description of the datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Nodes</th>
<th># Edges</th>
<th># Attributes</th>
<th># Time Steps</th>
<th># Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brain</td>
<td>5000</td>
<td>1955488</td>
<td>20</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>Reddit</td>
<td>8291</td>
<td>264050</td>
<td>20</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>DBLP-5</td>
<td>6606</td>
<td>42815</td>
<td>100</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>DBLP-3</td>
<td>4257</td>
<td>23540</td>
<td>100</td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

3.6 Computational Analysis

STAR is local in time and the length of input sequence does not affect its storage requirements. The complexity of STAR per time step is proportional to the number of parameters. The parameters of STAR are from the spatio-temporal GRU and the dual attention mechanism. Specifically, the number of parameters for generating the neighborhood vector and updating the spatio-temporal cell are

\[ O(d_d^2 + d_d m) \]

respectively. Because \( d_d = d_d \) and \( d_d > m \), the complexity of the spatio-temporal GRU per time step is

\[ O(K d_d^2 + d_d m) \]

The number of parameters of the dual attention is

\[ O(2 d_d^2 + d_d m) \]

Because \( d_d = d_d \) and \( d_d > m \), the complexity of the dual attention is

\[ O(K d_d^2 + d_d m) \]

Thus, the complexity per time step of STAR is

\[ O(K d_d^2 + d_d m) \]

4 Experiments

4.1 Datasets and Baselines

We use four real datasets as shown in Table 1. In the Brain dataset, the nodes represent the tidy cubes of brain tissue and the edges indicate the connectivity [Preti et al., 2017]. We apply PCA to the functional magnetic resonance imaging (fMRI) data\(^1\) to generate node attributes. Two nodes are connected if they show similar degree of activation [Gonzalez-Castillo et al., 2015]. In the Reddit dataset\(^2\), the nodes represent the posts. Two posts are connected if they contain similar keywords. We apply the word2vec approach [Mikolov et al., 2013] to the comments of a post to generate its node attributes [Hamilton et al., 2017]. The DBLP-3 and DBLP-5 datasets are extracted from the bibliography website DBLP\(^3\). Both of them are the co-author graphs where nodes represent authors. The paper titles and abstracts are used to generate node attributes by word2vec. The authors in DBLP-3 and DBLP-5 are from three and five research areas respectively.

We compare STAR with the state-of-the-art baseline methods. The comparison between them is shown in Table 2. DeepWalk [Perozzi et al., 2014], node2vec [Grover and Leskovec, 2016], GCN [Kipf and Welling, 2017] and GraphSAGE [Veličković et al., 2018] are originally designed for static graphs. They can not model the temporal patterns of the sequence data. We first apply them to each time step to generate an node representation. Then we concatenate the

\[ \text{https://tinyurl.com/y4hhw8ro} \]

\[ \text{https://www.reddit.com/} \]

\[ \text{https://dblp.uni-trier.de/} \]
representations of different time steps into a vector. The node labels are predicted by employing a logistic regression model on the vector. LSTM [Hochreiter and Schmidhuber, 1997], GRU [Cho et al., 2014], DynGEM [Goyal et al., 2018b] and DynAERNN [Goyal et al., 2018a] can model the temporal patterns. LSTM and GRU only consider the evolution of node attributes. DynGEM and DynAERNN can model the neighborhood information.

To evaluate the effectiveness of different components of STAR, we study its variants. STAR-NH is the variant that generates node representations only based on node attributes. The temporal attention is applied. STAR-TA is the variant without applying the temporal attention. STAR-SA is the variant without applying the spatial attention. It generates the aggregation of the neighborhood by the mean pooling.

In our experiments, $d_g$, $d_h$ and $d_a$ are set to 10. $K$ and $m$ are set to 2 and 3 respectively. $\lambda_1$ and $\lambda_2$ are set to $10^{-3}$. They are determined by grid-search from $\{1,2,5,10,15\}$, $\{1,2,3,4\}$ and $\{0,10^{-5},10^{-4},10^{-3},10^{-2}\}$ respectively. $N(\cdot)$ is the uniform sampling function. The number of sampled neighbors is set to 4. LeakyReLU [Maas et al., 2013] is adopted as $F(\cdot)$. We apply 10-fold cross-validation to evaluate all supervised methods. STAR is implemented with Tensorflow [Abadi et al., 2016] and optimized by Adam [Kingma and Ba, 2014]. The code of STAR and the data used are publicly available.

4.2 Experimental Results

Node classification. We evaluate the classification results by accuracy, AUC and F1 [Grover and Leskovec, 2016], which are shown in Table 3. It is observed that STAR achieves the best performance. DeepWalk and node2vec show good performance on the Reddit dataset. This is because the node labels of Reddit are largely dominated by the graph topology information, which is well utilized by DeepWalk and node2vec. LSTM and GRU show good performance on Brain, DBLP-3 and DBLP-5, where the node labels are dominated by node attributes information. STAR shows the best performance on all datasets, which indicates STAR utilizes well both graph topology and node attributes. Moreover, compared to DynGEM and DynAERNN, STAR shows a better ability of modeling the evolution patterns of temporal attributed graphs. STAR outperforms STAR-NH, indicating the importance of the spatial information of neighborhood on extracting better representations. By comparing STAR with STAR-TA and STAR-SA, we see the benefits of applying the dual attention mechanism.

Visualization of temporal attention. To investigate the interpretability of STAR, we visualize the attention values of different time steps. Figure 2 shows the results on four categories of nodes from the Brain dataset. It is observed that the nodes belonging to different categories show different kinds of attention distributions across time steps. This is because different categories of brain nodes show different degrees of activation when the subject conducts different tasks. For example, during time steps 7, 8 and 11, the subject mainly focuses on processing the audio information. As can be seen in Figure 2(a), the attention values of time steps 7, 8 and 11 are significantly larger for the nodes belonging to the category of auditory processing. Similar observations can be observed in Figures 2(b)-2(d). Larger values indicate that these time steps are more important for the node representations for classification. This demonstrates the effectiveness of the temporal attention mechanism. In addition, for Figure 2(a), the attention values of time steps 7, 8 and 11 are generated by three temporal attention units respectively. The same case exists for Figures 2(b)-2(d). This verifies the effectiveness of the multiple temporal attention mechanism.

Visualization of spatial attention. To gain further insight about the interpretability of STAR, we visualize the attention values of different neighbors. The results of two nodes from the Brain dataset, node 4023 and node 3266, are shown in Figure 3. Different node colors represent different node categories. Solid red and dotted gray lines represent higher and lower attention values respectively. Higher values indicate the neighbors are more important to the target node’s representation. As can be seen in Figure 3, the target node is influenced more by the neighbors from the same category. This is because the nodes from the same category play similar roles in the graph. For example, for node 4023, its immedi-

<table>
<thead>
<tr>
<th>Method</th>
<th>Brain ACC</th>
<th>Brain AUC</th>
<th>Brain F1</th>
<th>DBLP-3 ACC</th>
<th>DBLP-3 AUC</th>
<th>DBLP-3 F1</th>
<th>DBLP-5 ACC</th>
<th>DBLP-5 AUC</th>
<th>DBLP-5 F1</th>
<th>Reddit ACC</th>
<th>Reddit AUC</th>
<th>Reddit F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>71.4</td>
<td>97.2</td>
<td>70.2</td>
<td>49.7</td>
<td>60.1</td>
<td>50.5</td>
<td>35.4</td>
<td>61.0</td>
<td>26.9</td>
<td>47.5</td>
<td>71.9</td>
<td>46.8</td>
</tr>
<tr>
<td>node2vec</td>
<td>71.0</td>
<td>96.8</td>
<td>70.6</td>
<td>51.6</td>
<td>63.0</td>
<td>51.6</td>
<td>36.9</td>
<td>64.2</td>
<td>27.2</td>
<td>48.0</td>
<td>72.2</td>
<td>47.9</td>
</tr>
<tr>
<td>GCN</td>
<td>65.0</td>
<td>86.7</td>
<td>60.1</td>
<td>47.4</td>
<td>90.4</td>
<td>51.5</td>
<td>33.7</td>
<td>50.0</td>
<td>28.9</td>
<td>23.9</td>
<td>50.0</td>
<td>17.3</td>
</tr>
<tr>
<td>GraphSAGE</td>
<td>69.4</td>
<td>96.7</td>
<td>74.1</td>
<td>71.8</td>
<td>87.0</td>
<td>70.8</td>
<td>71.0</td>
<td>90.7</td>
<td>69.7</td>
<td>42.5</td>
<td>66.8</td>
<td>42.5</td>
</tr>
<tr>
<td>LSTM</td>
<td>83.6</td>
<td>98.6</td>
<td>84.6</td>
<td>81.9</td>
<td>92.5</td>
<td>81.7</td>
<td>74.1</td>
<td>91.4</td>
<td>74.1</td>
<td>40.2</td>
<td>66.5</td>
<td>40.6</td>
</tr>
<tr>
<td>GRU</td>
<td>81.6</td>
<td>98.6</td>
<td>82.2</td>
<td>82.5</td>
<td>93.7</td>
<td>83.2</td>
<td>75.6</td>
<td>91.5</td>
<td>75.2</td>
<td>42.1</td>
<td>67.2</td>
<td>41.9</td>
</tr>
<tr>
<td>DynGEM</td>
<td>71.0</td>
<td>97.2</td>
<td>70.2</td>
<td>52.3</td>
<td>59.0</td>
<td>52.8</td>
<td>31.6</td>
<td>54.6</td>
<td>9.9</td>
<td>39.9</td>
<td>66.2</td>
<td>41.5</td>
</tr>
<tr>
<td>DynAERNN</td>
<td>46.6</td>
<td>89.0</td>
<td>47.0</td>
<td>50.2</td>
<td>53.5</td>
<td>50.3</td>
<td>36.8</td>
<td>55.9</td>
<td>16.0</td>
<td>28.9</td>
<td>53.6</td>
<td>18.6</td>
</tr>
<tr>
<td>STAR-NH</td>
<td>84.7</td>
<td>98.4</td>
<td>86.1</td>
<td>83.1</td>
<td>94.4</td>
<td>83.5</td>
<td>76.6</td>
<td>92.2</td>
<td>75.9</td>
<td>42.3</td>
<td>67.1</td>
<td>42.1</td>
</tr>
<tr>
<td>STAR-TA</td>
<td>81.3</td>
<td>93.5</td>
<td>81.7</td>
<td>78.2</td>
<td>86.6</td>
<td>78.3</td>
<td>74.5</td>
<td>91.7</td>
<td>74.7</td>
<td>46.1</td>
<td>71.3</td>
<td>46.2</td>
</tr>
<tr>
<td>STAR-SA</td>
<td>79.5</td>
<td>90.2</td>
<td>79.9</td>
<td>78.3</td>
<td>86.5</td>
<td>79.6</td>
<td>72.1</td>
<td>88.5</td>
<td>72.6</td>
<td>44.6</td>
<td>68.0</td>
<td>44.4</td>
</tr>
<tr>
<td>STAR</td>
<td>89.2</td>
<td>99.2</td>
<td>90.0</td>
<td>86.2</td>
<td>97.1</td>
<td>86.7</td>
<td>80.3</td>
<td>95.5</td>
<td>80.7</td>
<td>50.8</td>
<td>75.0</td>
<td>51.1</td>
</tr>
</tbody>
</table>

Table 3: Node classification comparison (%)  

4https://tinyurl.com/y67ywq6j
In recent years, node classification has drawn extensive research attention [Perozzi et al., 2014; Grover and Leskovec, 2016; Wang et al., 2016; Kipf and Welling, 2017; Wang et al., 2017b; Hamilton et al., 2017]. Random walks are employed to extract the local information of a node in the graph. Nodes are further classified based on the local information preserved by word embedding techniques [Perozzi et al., 2014]. A biased random walk is proposed to learn richer node representations by exploring diverse local information [Grover and Leskovec, 2016]. Recently, some interest has been focused on the use of attention mechanism for node classification [Veličković et al., 2018]. These approaches are for the static graphs. Graphs in many real-world applications are dynamic. Both graph topology and node attributes evolve over time. Some recent progresses have been made on the node classification in the temporal graphs [Zhu et al., 2018; Du et al., 2018; Goyal et al., 2018a; Ma et al., 2018; Zuo et al., 2018; Yu et al., 2018]. The triadic closure process is used to model the evolution patterns of temporal graphs and a logistic regression model is used to classify nodes [Zhou et al., 2018]. To make the node representations stable over time, the node representation at time $t$ is generated from the one at time $t-1$ [Goyal et al., 2018b]. SLIDE [Li et al., 2018] uses a low-rank matrix to preserve all observed nodes in the graph where both structure and attributes change. However, none of these approaches can differentiate the relative importance of different factors that influence node representations.

5 Related Work

In recent years, node classification has drawn extensive research attention [Perozzi et al., 2014; Grover and Leskovec, 2016; Wang et al., 2016; Kipf and Welling, 2017; Wang et al., 2017b; Hamilton et al., 2017]. Random walks are employed to extract the local information of a node in the graph. Nodes are further classified based on the local information preserved by word embedding techniques [Perozzi et al., 2014]. A biased random walk is proposed to learn richer node representations by exploring diverse local information [Grover and Leskovec, 2016]. Recently, some interest has been focused on the use of attention mechanism for node classification [Veličković et al., 2018]. These approaches are for the static graphs. Graphs in many real-world applications are dynamic. Both graph topology and node attributes evolve over time. Some recent progresses have been made on the node classification in the temporal graphs [Zhu et al., 2018; Du et al., 2018; Goyal et al., 2018a; Ma et al., 2018; Zuo et al., 2018; Yu et al., 2018]. The triadic closure process is used to model the evolution patterns of temporal graphs and a logistic regression model is used to classify nodes [Zhou et al., 2018]. To make the node representations stable over time, the node representation at time $t$ is generated from the one at time $t-1$ [Goyal et al., 2018b]. SLIDE [Li et al., 2018] uses a low-rank matrix to preserve all observed nodes in the graph where both structure and attributes change. However, none of these approaches can differentiate the relative importance of different factors that influence node representations.

6 Conclusion

In this paper, we propose a novel method, STAR, for node classification in temporal attributed graphs. STAR consists of a spatio-temporal GRU and a dual attention module. By feeding the sequential node attributes and the neighborhood representations into the GRU, the temporal features of attributes evolution and the spatial information of the node’s local neighborhood can be effectively modeled respectively. A dual attention mechanism is developed to perform a thorough analysis on the interpretability of STAR. And it helps STAR detect the time steps and the node neighbors that are more important for the classification. Extensive experimental results demonstrate the effectiveness of STAR.

Acknowledgements

This work was partially supported by the National Science Foundation grant IIS-1707548.
References


