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

Dongkuan Xu†, Wei Cheng‡,
Dongsheng Luo†, Xiao Liu† and Xiang Zhang†
†The Pennsylvania State University ‡NEC Laboratories America, Inc.

Motivation

Node classification in graph-structured data aims to classify the nodes where the labels are only available for a subset of nodes. In real-world applications, both graph topology and node attributes evolve over time. Existing approaches, however, mainly focus on static graphs and lack the capability to simultaneously learn both temporal and spatial/structural features. Besides, 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.

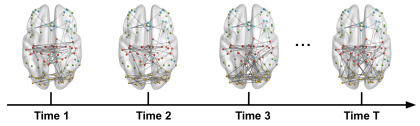
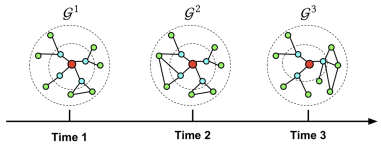


Fig. 1: Temporal brain graph.

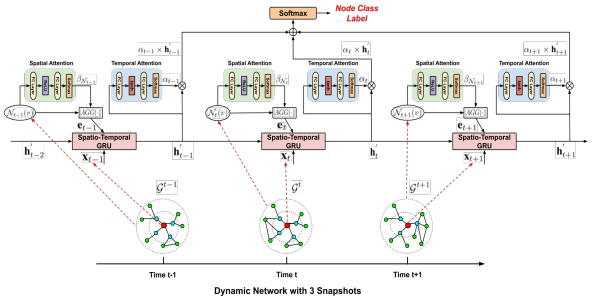
Problem



Goal: What Is The Category of The Red Node

- Notations
  - Temporal attributed graph $\mathbb{G} = (\mathcal{G}^1, \mathcal{G}^2, \cdots, \mathcal{G}^T)$, where $\mathcal{G}^t = (\mathcal{V}, \mathbf{A}^t, \mathbf{X}^t)$
    Node set $\mathcal{V}$, Adjacency matrix $\mathbf{A}^t \in \mathbb{R}^{N\times N}$, Node attributes $\mathbf{X}^t \in \mathbb{R}^{N\times d}$
- Definition
  - **Given:** $\mathbb{G}$ **and labels of a subset of nodes** $\mathcal{V}_L$
  - **Goal: to classify the nodes in subset** $\mathcal{V}_U$**, where** $\mathcal{V} = \mathcal{V}_L \cup \mathcal{V}_U$

Model Architecture



Dynamic Network with 3 Snapshots

Objective Function

Given the node representations $\mathbf{Q}_1, \cdots, \mathbf{Q}_N$ and the node labels $\mathbf{y}_1, \cdots, \mathbf{y}_N$, where N is the number of nodes, the objective function is

$$J = L_{ce} + \lambda_1 P_{att} + \lambda_2 P_{nn}. \tag{1}$$

$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N} \mathbf{y}_i \log(\tilde{\mathbf{y}}_i)$ is the cross-entropy loss. $\tilde{\mathbf{y}}_i = \text{softmax}(\mathbf{W}_o\mathbf{Q}_i + \mathbf{b}_o)$ is the estimate. c is the number of classes. $P_{att} = \|\mathbf{A}\mathbf{A}^\top - \mathbf{I}\|_F^2$ is the penalization term to encourage multiple attentions to diverge from each other. P_{nn} is the penalization term for over-fitting.

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. $\mathbf{e}_t$ is the vector representation of the neighborhood.

Standard GRU

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z[\mathbf{x}_t \oplus \mathbf{h}_{t-1}] + \mathbf{b}_z),\\
\mathbf{r}_t &= \sigma(\mathbf{W}_r[\mathbf{x}_t \oplus \mathbf{h}_{t-1}] + \mathbf{b}_r),\\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_h[\mathbf{x}_t \oplus (\mathbf{r}_t \odot \mathbf{h}_{t-1})] + \mathbf{b}_h),\\
\mathbf{h}_t &= (\mathbf{1} - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t,
\end{aligned}$$

v.s.

Spatio-Temporal GRU

$$\begin{aligned}
\mathbf{z}_t^{'} &= \sigma(\mathbf{W}_z^{'}[\mathbf{x}_t \oplus \mathbf{h}_{t-1}^{'} \oplus \mathbf{e}_t] + \mathbf{b}_z^{'}),\\
\mathbf{r}_t^{'} &= \sigma(\mathbf{W}_r^{'}[\mathbf{x}_t \oplus \mathbf{h}_{t-1}^{'} \oplus \mathbf{e}_t] + \mathbf{b}_r^{'}),\\
\mathbf{s}_t^{'} &= \sigma(\mathbf{W}_s^{'}[\mathbf{x}_t \oplus \mathbf{h}_{t-1}^{'} \oplus \mathbf{e}_t] + \mathbf{b}_s^{'}),\\
\tilde{\mathbf{h}}_t^{'} &= \tanh(\mathbf{W}_h^{'}[\mathbf{x}_t \oplus (\mathbf{r}_t^{'} \odot \mathbf{h}_{t-1}^{'}) \oplus (\mathbf{s}_t^{'} \odot \mathbf{e}_t)] + \mathbf{b}_h^{'}),\\
\mathbf{h}_t^{'} &= (\mathbf{1} - \mathbf{z}_t^{'}) \odot \mathbf{h}_{t-1}^{'} + \mathbf{z}_t^{'} \odot \tilde{\mathbf{h}}_t^{'},
\end{aligned}$$

Dual Attention Mechanism

A spatial attention module is designed to detect the important neighbors of a node. Based on the attention values, the aggregator sums up the neighbors' representations:

$$AGG_k(\{\mathbf{g}_{t(u)}^k, \forall u \in \mathcal{N}(v)\}) = \sum_{u\in\mathcal{N}(v)} \beta_u^k \mathbf{V}_k \mathbf{g}_{t(u)}^k, \tag{2}$$

The attention value β_u^k is produced as follows.

$$\beta_u^k = \frac{\exp\{F(\mathbf{w}_k^\top[\mathbf{V}_k\mathbf{g}_{t(u)}^k \oplus \mathbf{V}_k\mathbf{g}_{t(v)}^k])\}}{\sum_{v'\in\mathcal{N}(v)}\exp\{F(\mathbf{w}_k^\top[\mathbf{V}_k\mathbf{g}_{t(v')}^k \oplus \mathbf{V}_k\mathbf{g}_{t(v)}^k])\}}, \tag{3}$$

The temporal attention module outputs is described as follows.

$$\alpha_t = \frac{\exp\{\tilde{\mathbf{w}}^\top \tanh(\tilde{\mathbf{V}}\mathbf{h}_t^{'})\}}{\sum_{i=1}^{T}\exp\{\tilde{\mathbf{w}}^\top \tanh(\tilde{\mathbf{V}}\mathbf{h}_i^{'})\}}. \tag{4}$$

α_t indicates the importance of time step t for determining the target node's label compared to others. The attention values of all state vectors are $\boldsymbol{\alpha} = \text{softmax}(\tilde{\mathbf{w}}^\top \tanh(\tilde{\mathbf{V}}\mathbf{H}^\top)) \in \mathbb{R}^T$. $\mathbf{H} = [\mathbf{h}_1^{'} \oplus \cdots \oplus \mathbf{h}_T^{'}]$ is the concatenation of all state vectors. We apply multiple temporal attention units to focus on different parts. The resulting attention value matrix is $\mathbf{A} = \text{softmax}(\tilde{\mathbf{W}}^\top \tanh(\tilde{\mathbf{V}}\mathbf{H}^\top)) \in \mathbb{R}^{m\times T}$. The final node representation is denoted by

$$\mathbf{Q} = \mathbf{A}\mathbf{H} \in \mathbb{R}^{m\times d_h}. \tag{5}$$

Vector Representation of 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 K-hop neighbors.

Algorithm 1: Preparing K-hop Neighbors

Input: Temporal attributed graph $\mathbb{G} = (\mathcal{G}^1, \mathcal{G}^2, ..., \mathcal{G}^T)$, A set of input nodes $\mathcal{B}$, Depth K
Output: K-hop neighbors at different time steps, $\mathcal{B}_t^0 \dots \mathcal{B}_t^K$

```
for t = 1, ..., T do
    B_t^0 ← B
    for k = 1, ..., K do
        B_t^k ← B_t^{k-1}
        for v ∈ B_t^{k-1} do
            B_t^k ← B_t^k ∪ N(v)
```

Algorithm 2: Generating Neighborhood Vector

Input: Temporal attributed graph $\mathbb{G} = (\mathcal{G}^1, \mathcal{G}^2, ..., \mathcal{G}^T)$, K-hop neighbors $\mathcal{B}_t^0 \dots \mathcal{B}_t^K$, where $\mathcal{B}_t^0 = \mathcal{B}$
Output: Neighborhood vector $\mathbf{e}_{t(v)}$ for all $v \in \mathcal{B}_t^0$

```
for t = 1, ..., T do
    g_{t(v)}^K ← x_{t(v)}, ∀v ∈ B_t^K
    for k = K-1, ..., 1 do
        for v ∈ B_t^k do
            g_{N(t(v))}^{k+1} ← AGG_{k+1}({g_{t(u)}^{k+1}, ∀u ∈ N(v)})
            g_{t(v)}^k ← σ(W_trans^{k+1}[g_{t(v)}^{k+1} ⊕ g_{N(t(v))}^{k+1}])
    for v ∈ B_t^0 do
        e_{t(v)} ← AGG_1({g_{t(u)}^1, ∀u ∈ N(v)})
```

Experimental Results

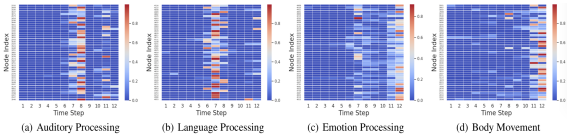
The comparison between the baseline methods is

Method	Models Temporal	Models Neighborhood	Handles Attribute	Applies Attention
DeepWalk	×	✓	×	×
node2vec	×	✓	×	×
GCN	×	✓	✓	×
GraphSAGE	×	✓	✓	×
LSTM	✓	×	✓	×
GRU	✓	×	✓	×
DynGEM	✓	✓	×	×
DynAERNN	✓	✓	×	×
STAR	✓	✓	✓	✓

We evaluate the classification results by accuracy, AUC and F1 shown as follows.

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

The attention values of different time steps for different categories of nodes from the Brain dataset are shown as follows.



(a) Auditory Processing (b) Language Processing (c) Emotion Processing (d) Body Movement

Contact

WELCOME! Email: dux19@psu.edu WeChat ID: xudongkuan220019