Draft Subreddit-County Temporal GNN Model

wpower

March 11 2021

1 GNN Update and Aggregation

$$h^k_{n,t} = f_{\text{Update}}(f^C_{\text{Agg}}(\mathbf{N}^{k-1}_{n,t}), f^S_{\text{Agg}}(\mathbf{S}^{k-1}_{n,t}), f^T_{\text{Agg}}(h^{k-1}_{n,t-1}, \cdots, h^{k-1}_{n,t-d}), h^{k-1}_{n,t}, x^{k-1}_{n,t}) \tag{1}$$

Where

f^C_{Agg} is the county-timestep-layer aggregation function.
f^S_{Agg} is the subreddit-layer aggregation function.
f^T_{Agg} is the temporal aggregation function.

The aggregation functions operate on the preceeding iterations messages, from the $(k-1)^{\text{th}}$ step of updates. The temporal aspect of the data is encoded in the parameters to the temporal temporal aggregation function. In this formaulation, there is a single, large graph with a node for each county, at each timestep. The temporal parameters represent the connections between different temporal realizations of a given county. In a sense, each node has a directional connection to d nodes 'in the past'. Note that the final parameter is the state of the node itself, these are the additional features for each county node, like the mobility counts per mobility vector.

The state of the node will be used to predict a COVID target through an additional function layer.

$$\hat{y}_{n,t} = f_{\text{Output}}\left(h_{n,t}\right)$$

These parameterized functions will then be trained by comparing this predicted output to the chosen COVID target.

1.1 $f^C_{\mathbf{Agg}}$

The parameters to this function represent the connections for a single county node n at a single timestep t. The set $\mathbf{N}^{k-1}_{n,t}$ contains the messages from the neighborhood of the county node n at the same timestep, from the preceeding round of updates.

A draft approach would be to use **symmetric normalization**, and sum the set of messages, each weighted by the factor $\frac{1}{\sqrt{\mathcal{N}_u * \mathcal{N}_v}}$

$$f^C_{\text{Agg}}(\mathbf{N}^{k-1}_{n,t}) = \sum_{v}^{\mathbf{N}} \left(\frac{h^{k-1}_{v,t}}{\sqrt{\mathcal{N}_v * \mathcal{N}_n}} \right)$$

This could be improved by the use of 'deep' methods, where a small parameterized function or network could be used to aggregate the messages.

1.2 $f^S_{\textbf{Agg}}$

The parameters to this function represent the connections between a single county-node n at a single timestep t and the set of 'static' subreddit nodes. The term static is used to refer to the fact that there is only a single node for a given subreddit, not a node for each timestep.
The set $\mathbf{S}^{k-1}_{n,t}$ represents the set of messages from all subreddits connected to the county at that timestep, from the preceeding iteration of the update function.
The message sent by a subreddit is simply a parameter holding its current 'embedding' in some low dimensional space, weighted by the edge weight (actual count value from the counties reddit activity for that timestep). For a subreddit s, its message to node n will be a function:

$$\mathbf{m}^k_{n,t,s} = h^{k-1}_s * g(e_{n,t,s})$$

Where $h^{k-1}_s \in \mathcal{R}^i$ is the state of the subreddit in the preceeding iteration of message passing, and i is the chosen dimension of the embedding space for subreddits. The function g weights this message by the value of the edge between the subreddit and the county node at that timestep, $e_{n,t,s}$
To aggregate these, the **Set Pooling** method would be used, where two MLPs are trained, with one operating directly on subreddit state values, and the second oeprating on the sum of these.

$$f^S_{\text{Agg}}(\mathbf{S}^{k-1}_{n,t}) = \text{MLP}_\theta \left(\sum_{s}^{S} \text{MLP}_\phi \left(\mathbf{m}^{k-1}_{n,t,s} \right) \right)$$

The weight values for the ϕ MLP would be shared across all subreddits.

1.3 $f^T_{\textbf{Agg}}$

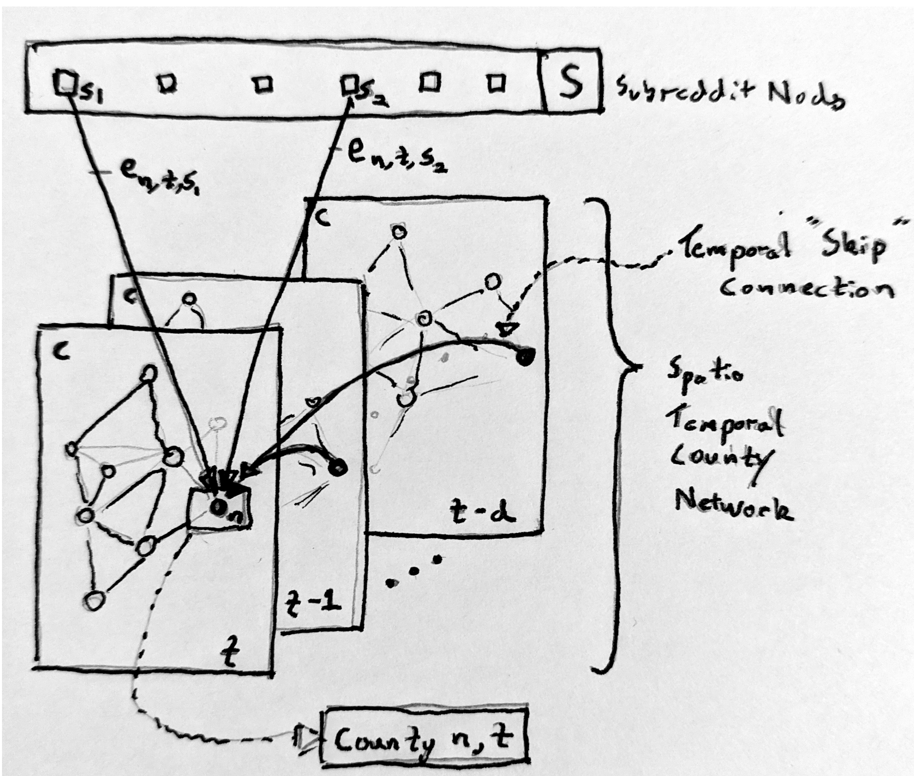
This function encodes the temporal history of a specific node, at a specific timestep. The sequence $x^{k-1}_{n,t-1}, \cdots, x^{k-1}_{n,t-d}$ represents the d nodes associated with county n, at the d preceeding timesteps.
To handle the sequential nature of this data, the a draft aggregation function could be an LSTM model.

$$f^T_{\text{Agg}}(h^{k-1}_{n,t-1}, \cdots, h^{k-1}_{n,t-d}) = \text{LSTM}(h^{k-1}_{n,t-1}, \cdots, h^{k-1}_{n,t-d})$$

In this example, the final hidden state could be used as the output of the aggregation function.

1.4 Message Overview

The following (poorly drawn) figure shows the sources of messages for a single county node, at a single time step, in a hypothetical county-subreddit network. Note the connections from the 'past' connecting the node to its d many historical instances. The node is also recieving messages from a subset of the subreddit layer.



1.5 $f_{\textbf{Update}}$

The final missing component is the update method that converts these signals into a new state for the county node. Note that only the county nodes will 'receive' an update. The subreddit nodes will be treated almost as parameters unto themselves, and won't update 'mid training' for the GNN, only when the GNN has stabalized, and the loss can be computed, will the 'state' of the subreddit nodes be updated.

The GRL book suggests naive approaches that simply apply a thin parameterized function over a concatenation of the aggregation outputs, the node features, and the prior state. This will need to be researched more, as the choice here will likely be critical.

$$f_{\text{Update}}(n,t) = \text{MLP}_v\left(\text{Concat}\left(f^C_{\text{Agg}}, f^S_{\text{Agg}}, f^T_{\text{Agg}}, h^{k-1}_{n,t}, x^{k-1}_{n,t}\right)\right)$$