Learning Expanding Graphs for Signal Interpolation

Bishwadeep Das and Elvin Isufi

B.Das@tudelft.nl
Introduction

• Typically, data Processing on graphs operations work with fixed-size graphs

• Graphs often grow in size

• This makes processing data over expanding graphs a challenge
Example: Recommendation Systems

- Graph filters process ratings over user graph to predict preferences for existing users (white cells of matrix)

- New user has no data, cannot attach to the user graph

1. Huang, W., et. al, Rating prediction via graph signal processing
All approaches rely on some information about the new node to operate, be it signal (Topology ID, Link Prediction), its connectivity (Link Prediction, related works).

Existing works on expanding graphs require incoming node connectivity\(^2\),\(^3\), or estimate it from features\(^4\).

**Gap:** Find a way to figure connectivity and subsequent data-processing for new nodes approaching a graph when no information is available.

---

2. Shen et. al., *Online Graph Adaptive Learning With Scalability and Privacy*
3. Venkitaraman et. al., *Recursive Prediction of Graph Signals With Incoming Nodes*
4. Dornaika et. al., *Efficient dynamic graph construction for inductive semi-supervised learning*
Problem Formulation

We consider a stochastic attachment model\textsuperscript{5,6}

\[
\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A} \in \mathbb{R}^{N\times N}, \mathbf{x} \in \mathbb{R}^N \} \quad \mathcal{G}_+ = \{\mathcal{V}_+, \mathcal{E}_+, \mathbf{A}_+ \in \mathbb{R}^{N+1\times N+1}, \mathbf{x}_+ \in \mathbb{R}^{N+1} \}
\]

- Node \( v_+ \) attaches to \( v_i \) with probability \( p_i \) and edge weight \( w_i \).
- Edges directed towards \( v_+ \).
- Attachment vector \( \mathbf{a}_+ \in \mathbb{R}^N, [\mathbf{a}_+]_i = w_i \) with prob. \( p_i \).

5. Erdos, P. and Rényi, A., On the evolution of random graphs
6. Barabási, A.L. and Albert, R., Emergence of scaling in random networks
Prob. Formulation (contd.)

• Expanded adjacency matrix:  

\[
A_+ = \begin{bmatrix}
A & 0 \\
\mathbf{a}_+^\top & 0
\end{bmatrix}
\]

• \(v_+\) has signal \(x_+\), we have the expanded signal \(x_+ = [x, x_+]^\top\)

• \(\mathbf{a}_+\) is an element-wise independent weighted Bernoulli random vector

• Its expectation is \(\mathbb{E}[\mathbf{a}_+] = w \cdot p\) and covariance \(\Sigma_+ = \text{diag}(w^2 \cdot p \cdot (1 - p))\)

• The adj. matrix after attachment obeys \(\mathbb{E}[A_+] = \begin{bmatrix}
A & 0 \\
(p \cdot w)^\top & 0
\end{bmatrix}\)
Adaption to a task

• Main task is to solve the parameters $\mathbf{w}, \mathbf{p}$ relative to a task

• Use training set $\mathcal{F} = \{(v_{t+}, x_{t+}, a_{t+}, b_{t+})\}_t$ for empirical risk minimisation

• $v_{t+}$: $t$-th node sample, $x_{t+}$: incoming node signal/label

• $a_{t+}$: sample attachment pattern, $b_{t+}$: binary sample attachment pattern

1. $v_+$
2. $x_+ = $
3. $a_+ = [0.3, 0, 0, 0, 0, 0, 0.4, 0, 0, 0]^{\top}$
4. $b_+ = [1, 0, 0, 0, 0, 0, 1, 0, 0, 0]^{\top}$
Adaption to a task

- Task-specific cost $f_{\mathcal{T}}(p, w, x_{t+})$

We solve

$$\min_{p, w} \mathbb{E}[f_{\mathcal{T}}(p, w, x_{t+})] + g_{\mathcal{T}}(p, b_{t+}) + h_{\mathcal{T}}(w, a_{t+})$$

subject to  $p \in [0, 1]^N$, $w \in \mathcal{W}$

$g_{\mathcal{T}}(\cdot), h_{\mathcal{T}}(\cdot)$ act as regularisers, $\mathcal{W}$ : constraint set for edge weights
Task: Interpolation at incoming node

- Predict signal at an incoming node with no prior information

- Node attaches to $\mathcal{G}$, expanded signal $x_+ = [x, 0]^T$ before interpolation

- For interpolation we use FIR graph filters\textsuperscript{7} with shift operator $A_+$

  - Filter Output $y_+ = \sum_{l=1}^{L} h_l A_+^l x_+$, filter $h = [h_1, \ldots, h_L]^T$

- Interested in the error $\mathbb{E}[(y_{N+1}^+ - x_+)^2]$
Task: Interpolation at incoming node

The MSE is

$$\text{MSE}(p, w) = \| (w \circ p)^T A_x h - x^*_+ \|^2_2 + h^T A_x^T \Sigma_+ A_x h$$

- Here, $$((w \circ p)^T A_x h - x^*_+)^2$$ is the bias for that node.

- The term $$h^T A_x^T \Sigma_+ A_x h$$ is the output variance.

- We need to avoid solutions like $$p = 1_N, 0_N$$ by using regularisers.
Training

\[
\min_{p, w} \text{MSE}_T(p, w) + \sum_{t=1}^{\mathcal{T}} \left( \mu_p \|p - b_{t+}\|_2^2 + \mu_w \|w - a_{t+}\|_2^2 \right)
\]

subject to \( p \in [0, 1]^N, w \in \mathcal{W} \)

- Not always convex in \( p \), convex in \( w \)
- We use alternating projected gradient descent

Convex in \( p \) when \( \mu_p \geq w_h^2 \max_{i \in \{1, \ldots, N\}} ([A_x h]_i)^2 - \|w \circ A_x h\|_2^2 \)
Numerical Results: Synthetic Graphs

- Erdos-Rényi and Barabasi-Albert, each of 100 nodes
- Generate band-limited graph signal
- Generate $\mathcal{T}$ with corresponding $p$, $w$ pair
- Use as filter the simple shift operator to generate $x_+$ at each node
- Evaluate MSE over 100 such realisations for each node
- Compare with uniformly random and preferential attachment
Numerical Results: Convergence

Training with learning rates $10^{-5}$

Ensuring marginal convexity not a good idea.
Numerical: MSE at incoming node

- Proposed outperforms rest, shows importance of task-data-topology coupling
- We also train separately for each variable, given the other
- Training only over $w$ performs better because of convexity in it
 Numerical Results: Item cold start collaborative filtering

Movielens 100K: 943 users, 1152 Items

<table>
<thead>
<tr>
<th></th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Item 2</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>Item 1</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Ratings Matrix

We predict ratings for new items for each user graph
Numerical Results: Violin plots

We do best in predicting ratings for new items in data scarcity settings.

Does better than other attachments.

Shows advantage of personalised recommendations.
Conclusion

- Data, topology and task-driven attachment model for incoming nodes without prior information
- Parameterised by attachment probabilities and edge-weights, obtained by alternating projected gradient descent
- Outperforms stochastic and purely data-driven attachment

Future Work

- Process a sequence of incoming nodes without repeated re-training.
- Processing data on both the existing graph and the incoming node.
Thanks