Relative and Absolute Location Embedding for Few-Shot Node Classification on Graph

Zemin Liu¹, Yuan Fang¹, Chenghao Liu¹,², Steven C.H. Hoi¹,²

¹ Singapore Management University, Singapore; ² Salesforce Research Asia, Singapore

In Proceeding of 35th AAAI Conference on Artificial Intelligence, February 2-9, 2021
Outline

• Problem definition & related work
• Challenges & insights
• Proposed model: RALE
• Experiments
• Conclusions
Problem definition: few-shot node classification

- **Base classes**
  - SVM
  - Neural networks

- **Novel classes**
  - Explainable AI
  - Fair ML

**Task:** few-shot node classification for novel classes.

Figure 1: Illustration of few-shot node classification.
Related work: graph representation learning

- **Graph embedding**
  - Capture local structure
  - *e.g.*, Deepwalk [1], node2vec [2]

- **Graph neural networks**
  - Recursive neighborhood aggregation
  - *e.g.*, GCN [4], GAT [5]

Related work: few-shot node classification

• Few-shot node classification
  – e.g., Meta-GNN [1], GFL [2]
  – Follow meta-learning paradigm
  – Formulate few-shot node classification as a series of classification tasks
  – Learn a transferable prior from meta-train tasks
  – Adapt the prior on meta-test tasks

• What is missing?
  – Graph nodes related to each other and the inter-dependency is crucial to learning node representations
  – These models overlook such inter-dependency in a task, which should be incorporated into the transferable prior

Outline

- Problem definition & related work
- Challenges & insights
- Proposed model: RALE
- Experiments
- Conclusions
Challenges

• Goal
  – Exploit additional dependencies based on graph structures to enhance the limited support set.

• Challenges
  – How to capture the potentially long-ranged dependencies between nodes within a task?
  – How to align the dependencies across tasks to converge on a common transferable prior?
Insights

• Hub nodes
  – Structurally important nodes
    • e.g., as measured by network centrality scores such as degree or PageRank [1]
    • Backbone of the graph
  – Two roles:
    • Utilize hubs for efficient sampling of important paths; relative location
    • Utilize hubs as a set of global references, to align the dependencies across tasks; absolute location

Figure 2: Hub-based relative and absolute locations.

Outline

• Problem definition & related work
• Challenges & insights
• Proposed model: RALE
• Experiments
• Conclusions
Proposed model: Overview of RALE

Abstraction of Location Embedding

- **Graph encoder** $\phi_g$: embed nodes into representations
- **Path encoder** $\phi_p$: embed paths into representations
- **Location embedding** for node $v$ w.r.t. reference nodes $\mathcal{R}$

\[
e^v_{\{u\}} = \text{AGGR}(\{\phi_p(\phi_g(p; \theta_g); \theta_p) : \forall p \in \mathcal{P}_{u,v}\})
\]

\[
e^\mathcal{R} = \phi(\{\mathcal{P}_{u,v} : u \in \mathcal{R}\}; \theta_g, \theta_p)
\]

\[
= \text{AGGR}(\{e^v_{\{u\}} : \forall u \in \mathcal{R}\}).
\]
Task- and graph-level location embedding

- **Task-level** relative location (task $t = (S_t, Q_t)$)
  \[
e_{v}^{S_t} = \phi(\{P_{s,v} : s \in S_t\}; \theta_g, \theta_p)\]

- **Graph-level** absolute location (hubs set $\mathcal{H}$)
  \[
e_{v}^{\mathcal{H}} = \phi(\{P_{h,v} : h \in \mathcal{H}\}; \theta_g, \theta_p)\]

Figure 2: Hub-based relative and absolute locations.
Dependency-aware meta-optimization

- Dependency-aware classification layer

\[ \psi(v; \Theta) = \text{SOFTMAX} \left( \sigma(\mathbf{W}[h_v||e^{S_t}_v||e^H_v]) \right) \]

- Meta-objective (MAML) [1]

\[ L(S_t; \Theta) = - \sum_{v \in S_t} \sum_{i=1}^{m} \mathbb{I}_{l(v)=i} \ln(\psi(v; \Theta)[i]) \]  

\[ \Theta' = \Theta - \alpha \frac{\partial L(S_t; \Theta)}{\partial \Theta} \]  

\[ \Theta^* = \arg \min_\Theta \sum_{t \in T_{tr}} L(Q_t; \Theta - \alpha \frac{\partial L(S_t; \Theta)}{\partial \Theta}) \]

Outline

• Problem definition & related work
• Challenges & insights
• Proposed model: RALE
• Experiments
• Conclusions
Datasets, evaluation and baselines

• Datasets

<table>
<thead>
<tr>
<th></th>
<th>Nodes</th>
<th>Edges</th>
<th>Features</th>
<th>Classes (Train/Val/Test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>13,381</td>
<td>245,778</td>
<td>767</td>
<td>10 (5/2/3)</td>
</tr>
<tr>
<td>Email</td>
<td>909</td>
<td>13,733</td>
<td>128</td>
<td>28 (15/6/7)</td>
</tr>
<tr>
<td>Reddit</td>
<td>231,371</td>
<td>11,606,876</td>
<td>602</td>
<td>41 (25/6/10)</td>
</tr>
</tbody>
</table>

• Baselines

**GNNs**: GCN, GraphSAGE, GAT
- Training: optimize GNN on all base classes
- Test: freeze parameters of aggregation layer, update parameters of classification layer

**GNN+’s**: GCN+, GraphSAGE+, GAT+
- Training: optimize GNN on all base classes
- Test: update all parameters

**Meta-learning models**: Meta-GNN, Proto-GNN

Empirical results

Table 2: Accuracy (percent) of RALE and baselines. In each row, the best result is bolded and the second best is underlined. RALE’s improvement is calculated relative to the best baseline, with the corresponding p-value under two-tail paired t-test.

<table>
<thead>
<tr>
<th></th>
<th>GCN</th>
<th>GraphSAGE</th>
<th>GAT</th>
<th>GCN+</th>
<th>GraphSAGE+</th>
<th>GAT+</th>
<th>Meta-GNN</th>
<th>Proto-GNN</th>
<th>RALE</th>
<th>impr.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-shot</td>
<td>71.97</td>
<td>63.19</td>
<td>72.61</td>
<td>70.86</td>
<td>68.36</td>
<td>66.99</td>
<td>73.20</td>
<td>68.91</td>
<td><strong>78.07</strong></td>
<td>+6.65%</td>
<td>0.007</td>
</tr>
<tr>
<td>3-shot</td>
<td>78.67</td>
<td>63.15</td>
<td>78.18</td>
<td>72.66</td>
<td>69.75</td>
<td>76.07</td>
<td>74.45</td>
<td>76.41</td>
<td><strong>84.17</strong></td>
<td>+6.99%</td>
<td>0.021</td>
</tr>
<tr>
<td>5-shot</td>
<td>79.46</td>
<td>64.71</td>
<td>85.18</td>
<td>79.84</td>
<td>68.44</td>
<td><strong>85.45</strong></td>
<td>76.58</td>
<td>79.67</td>
<td>84.50</td>
<td>-1.11%</td>
<td>0.437</td>
</tr>
<tr>
<td>Email</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-shot</td>
<td>43.15</td>
<td>40.25</td>
<td>43.11</td>
<td>47.15</td>
<td>43.48</td>
<td>40.82</td>
<td>47.21</td>
<td>35.47</td>
<td><strong>51.82</strong></td>
<td>+9.76%</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>3-shot</td>
<td>53.29</td>
<td>45.56</td>
<td>41.06</td>
<td>50.51</td>
<td>44.28</td>
<td>35.83</td>
<td>55.64</td>
<td>40.93</td>
<td><strong>59.47</strong></td>
<td>+6.88%</td>
<td>0.002</td>
</tr>
<tr>
<td>5-shot</td>
<td>58.19</td>
<td>48.38</td>
<td>37.24</td>
<td>58.37</td>
<td>47.01</td>
<td>33.52</td>
<td>58.76</td>
<td>42.38</td>
<td><strong>65.46</strong></td>
<td>+11.40%</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>Reddit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-shot</td>
<td>20.02</td>
<td>41.61</td>
<td>19.95</td>
<td>20.13</td>
<td>35.89</td>
<td>19.99</td>
<td>46.42</td>
<td>44.95</td>
<td><strong>48.63</strong></td>
<td>+4.76%</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>3-shot</td>
<td>20.16</td>
<td>47.00</td>
<td>20.15</td>
<td>20.21</td>
<td>49.11</td>
<td>20.02</td>
<td>51.28</td>
<td>51.60</td>
<td><strong>56.85</strong></td>
<td>+10.17%</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>5-shot</td>
<td>20.40</td>
<td>50.53</td>
<td>20.12</td>
<td>20.41</td>
<td>51.30</td>
<td>20.14</td>
<td>53.33</td>
<td>52.57</td>
<td><strong>57.45</strong></td>
<td>+7.73%</td>
<td>0.029</td>
</tr>
</tbody>
</table>

**Observations**

- GNNs vs GNN+’s: no consistent winner emerge
- Meta-GNN and Proto-GNN achieve comparable or slightly better performance than GNNs and GNN+’s
- RALE generally outperforms all baselines
- More shots for support set
  - GNNs and GNN+’s: fluctuate
  - Meta-learning methods: steadily perform better
Ablation study and parameter analysis

- **Ablation study**
  - RALE\ar has the lowest accuracy
  - RALE\r and RALE\a perform better than RALE\ar; RALE\a is slightly better than RALE\r.
  - Full model RALE achieves the best performance

- **Parameters sensitivity**
  - $d = 32$ appears to be robust across datasets
  - $\alpha \in [0.1, 10]$ is better, though they may have different optimal values due to the various relationships between tasks
Outline

- Problem definition & related work
- Challenges & insights
- Proposed model: RALE
- Experiments
- Conclusions
Conclusions

• Problem
  – Few-shot node classification on graph

• Motivation
  – Nodes on a graph are non-i.i.d
  – Need to capture the node dependencies within tasks

• Proposed model: RALE
  – hub-based relative and absolute location embedding
    • Relative locations capture the task-level dependency
    • Absolute locations capture the graph-level dependency to align the tasks
Thanks!

Zemin Liu, Yuan Fang, Chenghao Liu, Steven C.H. Hoi.

Relative and Absolute Location Embedding for Few-Shot Node Classification on Graph.

In Proceeding of 35th AAAI Conference on Artificial Intelligence, February 2-9, 2021