Self-Supervised Learning of Contextual Embeddings for Link Prediction in Heterogeneous Networks

Ping Wang¹, Khushbu Agarwal², Colby Ham², Sutanay Choudhury², Chandan K. Reddy¹

1. Department of Computer Science, Virginia Tech, Arlington, VA
2. Pacific Northwest National Laboratory, Richland, WA
Outline

- Introduction
- Challenges
- Our Contributions
- Experiments
- Conclusion
Heterogeneous networks:
- Integrate different data sources, build relations between them.
- Allows us to further discover underlying correlations with link prediction task.

Existing link prediction methods:
- Provide a static embedding for each entity that is agnostic to any specific context.
- Learn the importance of all the node’s immediate and multi-hop neighbors in the graph.
- Without considering contextual information of the downstream task.

Motivation of Contextual Embeddings

Academic Network with authors publish on diverse topics

State-of-the-art methods aggregate global semantics for authors based on all published papers

Our approach uses context subgraph between authors to contextualize their node embeddings
Existing Challenges

- How to characterize the context in heterogeneous network?
  - Existing contextual embedding learning:
    - Define a node’s participation in different contexts as facets or aspects.
    - Infer the cluster-driven context for nodes in the downstream tasks.
    - However, accounting for higher-order effects over diverse meta-paths or meta-graphs is demonstrated to be essential in heterogeneous networks.

- How to learn the contextual node embeddings according to different contexts?
- How to show the effectiveness of contextual embedding and interpret the results?
Our Contributions

We developed a framework bridging static representation learning methods using global information from the entire graph with localized subgraph-based context to learn contextual node representations.

**Define Contextual Subgraphs**
- Contextual embeddings are learnt based on task-specific subgraphs.
- Node representations will be dynamically changing with different subgraphs.

**Self-supervised Learning Approach**
- Learn higher-order semantic associations by simultaneously capturing the global information and local context.
- Two training stages: pre-training and fine-turning.

**Performance Evaluation**
- Compare with static and contextual embedding learning methods.
- Demonstrate the effectiveness and interpretability of contextual embeddings.
SLiCE Framework

Context graph gc for Authors A1, A2 (indicated by grey edges)

Subgraph Based Context Generation

Context Encoder

Node global embedding
Node type embedding

Contextual Translation

Global and Local Aggregation

Semantic association for A1 changes from high attention in global associations to high attention in local context subgraph, over the contextual translation layers.

Pretraining

Link prediction score

Mask and predict nodes

Fine-Tuning

Node global embedding
Node type embedding
SLiCE: Context Generation and Representation

- **Context Generation**: generate context for each node or a node-pair.
  - Random strategy: BFS based star graph; random walks with a certain depth.
  - Shortest Path: consider the shortest path between two nodes.

- **Context Representation**:
  - Subgraph $g_c$ is encoded as $g_c = (v_1, v_2, ..., v_{|V_c|})$. Here, $|V_c|$ is the number of nodes in $g_c$.
  - Global embeddings of nodes in $g_c$ are represented as $H_c = (h_1, h_2, ..., h_{|V_c|})$. Where, $h_i$ is the low-dimensional representation of node $i$ that considers various information in the global graph, such as the structures and attributes.
  - We mainly consider the pre-trained node embeddings from node2vec, which is a random walk based skip-gram methods.
SLiCE: Contextual Translation

- Semantic association matrix $\tilde{A}$:
  - Given two nodes $v_i$ and $v_j$ in the context, the corresponding entry $\tilde{A}_{ij}^k$ can be computed as follows.
  \[
  \frac{\exp\left((w_1 h_i^k)^T (w_2 h_j^k)\right)}{\sum_{t=1}^{V_c} \exp\left((w_1 h_t^k)^T (w_2 h_t^k)\right)}
  \]

- Contextual Translation: Apply multiple translation layers; in $k + 1$ layer, $\tilde{A}^k$ is updated as follows:
  \[
  H_{c}^{k+1} = f_{NN} (W_s H_c^k \tilde{A}^k + H_c^k)
  \]

- The node embeddings from different layers ($K$ in total) are aggregated as the contextual embedding:
  \[
  \tilde{h}_i = h_1^1 \oplus h_2^2 \oplus \cdots \oplus h_i^K
  \]
SLiCE: Contextual Embedding Tasks

- **Self-supervised Contextual Node Prediction in Pre-training:**
  - Generate context for each node in the network, and randomly mask a node for prediction.
  - Objective: maximize the probability of observing the masked node based on the context.

- **Fine-tuning with Supervised Contextual Link Prediction:**
  - Generate context for each node-pair and perform the binary link prediction.
  - Objective: maximizing the prediction score of positive edges and minimizing the score for negative edges.
  - The probability of the edge between two nodes is calculated as the similarity score between their contextual embeddings.
Datasets used:

- Amazon (E-commerce): co-viewing and co-purchasing links between products.
- DBLP (Academic): relationships between papers, authors, venues and terms.
- Freebase (Knowledge Base): links between people and their demographic features.
- Twitter (Social Networks): links between tweets users.
- Healthcare: relations between patients and their diagnosed medical conditions, procedures and medications received during each hospital admission.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Amazon</th>
<th>DBLP</th>
<th>Freebase</th>
<th>Twitter</th>
<th>Healthcare</th>
</tr>
</thead>
<tbody>
<tr>
<td># Nodes</td>
<td>10,099</td>
<td>37,791</td>
<td>14,541</td>
<td>9,990</td>
<td>4,683</td>
</tr>
<tr>
<td># Edges</td>
<td>129,811</td>
<td>170,794</td>
<td>248,611</td>
<td>294,330</td>
<td>205,428</td>
</tr>
<tr>
<td># Relations</td>
<td>2</td>
<td>3</td>
<td>237</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td># Training (positive)</td>
<td>126,535</td>
<td>119,554</td>
<td>272,115</td>
<td>282,115</td>
<td>164,816</td>
</tr>
<tr>
<td># Development</td>
<td>14,756</td>
<td>51,242</td>
<td>35,070</td>
<td>32,926</td>
<td>40,612</td>
</tr>
<tr>
<td># Testing</td>
<td>29,492</td>
<td>51,238</td>
<td>40,932</td>
<td>65,838</td>
<td>40,612</td>
</tr>
</tbody>
</table>

1. Codes for generating the Healthcare network based on MIMIC III is available at [https://github.com/pnnl/SLICE](https://github.com/pnnl/SLICE)
How SLiCE works?

<table>
<thead>
<tr>
<th>Type</th>
<th>Methods</th>
<th>micro-F1 score</th>
<th>AUCROC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Amazon</td>
<td>DBLP</td>
</tr>
<tr>
<td>Static</td>
<td>TransE</td>
<td>50.28</td>
<td>49.60</td>
</tr>
<tr>
<td></td>
<td>RefE</td>
<td>51.86</td>
<td>49.60</td>
</tr>
<tr>
<td></td>
<td>node2vec</td>
<td>88.06</td>
<td>86.71</td>
</tr>
<tr>
<td></td>
<td>metapath2vec</td>
<td>88.86</td>
<td>44.58</td>
</tr>
<tr>
<td>Contextual</td>
<td>GAN</td>
<td>85.47</td>
<td>OOM</td>
</tr>
<tr>
<td></td>
<td>GATNE-T</td>
<td>89.06</td>
<td>57.04</td>
</tr>
<tr>
<td></td>
<td>RGCN</td>
<td>65.03</td>
<td>28.84</td>
</tr>
<tr>
<td></td>
<td>CompGCN</td>
<td>83.42</td>
<td>40.10</td>
</tr>
<tr>
<td></td>
<td>HGT</td>
<td>65.77</td>
<td>53.32</td>
</tr>
<tr>
<td></td>
<td>asp2vec</td>
<td>94.89</td>
<td>78.82</td>
</tr>
<tr>
<td></td>
<td>SLiCE w/o GF</td>
<td>67.01</td>
<td>66.02</td>
</tr>
<tr>
<td></td>
<td>SLiCE w/o ET</td>
<td>94.99</td>
<td>89.34</td>
</tr>
<tr>
<td></td>
<td>SLiCE (Ours)</td>
<td>96.00*</td>
<td>90.70*</td>
</tr>
</tbody>
</table>

• The symbol “OOM” indicates out of memory.
• The symbol * indicates that the improvement is statistically significant over the best baseline by two-sided t-test with p-value $10^{-10}$. 
Where Does the Context Help?

Similar results are obtained on Amazon, Freebase and Twitter datasets.
Nodes in the context from DBLP:

- **Paper N0**: Summarizing itemset patterns: a profile-based approach
- **Author N1**: Jiawei Han
- **Topic N2**: Approach
- **Paper N3**: Knowledge-base reduction
- **Topic N4**: Redundancy
- **Paper N5**: Handling Redundancy in the Processing of Recursive Database Queries
How to Interpret the Prediction on DBLP?

Prediction Scores:
1: 0.9036
2: 0.9118
3: 0.8875
4: 0.9642
5: 0.8208

- **P28406**: Summarizing itemset patterns: a profile-based approach.
- **P12588**: Knowledge-Base Reduction: A New Approach to Checking knowledge Bases for Inconsistency and Redundancy.
- **P12802**: Handling Redundancy in the Processing of Recursive Database Queries.
- **P33963**: CloSpan: Mining Closed Sequential Patterns in Large Databases.
- **P8201**: Association Mining in Large Databases: A Re-examination of Its Measures.
- **P9072**: SpaRClus: Spatial Relationship Pattern-Based Hierarchial Clustering.
- **P32528**: Star-Cubing: Computing Iceberg Cubes by Top-Down and Bottom-Up Integration.
- **P30219**: ARCube: supporting ranking aggregate queries in partially materialized data cubes.
Conclusions

- Developed a contextual embedding learning method for graphs from single relation context to arbitrary subgraphs.

- Our self-supervised learning based method can model higher-order semantic associations between nodes.

- SLiCE contextual embeddings significantly outperform existing embedding learning methods for link prediction in heterogeneous networks.

- SLiCE model results are interpretable, effective, and scalable.
Thank You!

Link to SLiCE implementation:
https://github.com/pnnl/SLICE

Feel free to send questions and suggestions to
Ping Wang
Email: ping@vt.edu
Web: https://people.cs.vt.edu/ping/