MARU: Representation Learning for Heterogeneous Networks with Meta-context Aware Random Walks

Jyun-Yu Jiang, Zeyu Li, Chelsea J.-T. Ju and Wei Wang

Scalable Analytics Institute (ScAi)
University of California, Los Angeles (UCLA)

CIKM 2020
Information networks are ubiquitous in our lives.

Social Networks

Medical Networks

Bibliographic Networks
Network Embedding for Machine Learning Approaches

Low-dimensional feature vectors can be directly applied on applications.
State-of-the-art: Learning from Random Walks

Many recent researches focus on generating “good” random walks.
Heterogeneous networks are more intuitive and general.

Medical Networks

Bibliographic Networks

Nodes can belong types with different semantic meanings.
Existing methods are limited for heterogeneous networks.

- Nodes can have **different frequencies** and distributions.
  - Popular nodes can be over-optimized.

- Node **types** are ignored.
  - Random walks with varied types can have different semantics.

- Relying on **prior knowledge and given strategies**.
  - Predefined meta-paths, i.e., short type sequences.
  - Ad-hoc random walk strategies.
  - Random walks are manipulated and limited.

A robust method for learning heterogeneous network representations is needed!
Motivation: Dynamically Learning with Meta-context

Different contexts of types should be modeled separately.

The number of feasible meta-contexts are usually limited!
MARU: The Proposed Framework

Representation Learning with Meta-context Aware Random Walks (MARU)

Bidirectional Extended Random Walk

Meta-context Aware Node Embedding

Meta-context Aware Skip-gram

Embedding Inference

final node representation $\Phi(v_i)$

$w_1 = P(m_1 | v_i)$

$w_{|M|} = P(m_{|M|} | v_i)$

$w = P(m | v_i)$
Bidirectional Extended Random Walk

Classical Random Walks

Bidirectional Extended Random Walks

Missing Nodes in Sliding Windows

Full Samples of Contexts

Tail nodes can also be theoretically better optimized.
Theoretical Proof for Bidirectional Extended Random Walk

**Corollary**

Assume head nodes are never transitioning to tail nodes in random walks, and the probability of transitioning between tail nodes is $p$. Given a tail starting node $u$ and the walking length $2n + 1$, the expected number of tail nodes in a bidirectional random walk is greater than the expected number in a one-directional random walk.

**Proof.**

For the one-directional random walk, the expected number of visited tail nodes is $E_o = 1 + \sum_{i=0}^{2n} i \cdot p^i \cdot (1 - p)$. For the bidirectional random walk, the expected number of visited tail nodes is $E_b = 1 + 2 \cdot \sum_{i=0}^{n} i \cdot p^i \cdot (1 - p)$. Therefore, we have

$$
\lim_{n \to \infty} E_b - E_o = \lim_{n \to \infty} (1 - p) \cdot \left( \sum_{i=1}^{n} i \cdot p^i - (i + n) \cdot p^{i+n} \right) > 0
$$
A node $v$ can have distinct representations $\Phi(v, m)$ for different meta-contexts $m$. 

**Bidirectional Extended Random Walk**

**Meta-context Aware Node Embedding**

**Meta-context Aware Skip-gram**

![Diagram](https://example.com/diagram.png)
Embedding Inference

Each node still needs an ultimate representation for applications.

Meta-context Aware Node Embedding

\[
\Phi(v_i, m_i) \rightarrow \Phi(v_i, m_1) \quad w_1 = P(m_1 | v_i)
\]

\[
\Phi(v_i, m_2) \rightarrow \Phi(v_i) \text{ final node representation}
\]

\[
\Phi(v_i, m_{|M|}) \rightarrow w_{|M|} = P(m_{|M|} | v_i)
\]

Embedding Inference
Three Evaluation Tasks
- Multi-label Node Classification (Supervised Learning)
- Node Clustering (Unsupervised Learning)
- Link Prediction (Supervised Learning)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Node Types and Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBIS</td>
<td>Author (A) 60,694 Paper (P) 72,902 Venue (V) 464</td>
</tr>
<tr>
<td></td>
<td>(264,323 edges)</td>
</tr>
<tr>
<td>MovieLens</td>
<td>Movie (M) 10,197 Actor (A) 95,321 Director (D) 4,060 User (U) 2,113</td>
</tr>
<tr>
<td></td>
<td>(1,097,495 edges)</td>
</tr>
<tr>
<td>Yelp</td>
<td>User (U) 16,239 Business (B) 14,284 Category (C) 511 Location (L) 47</td>
</tr>
<tr>
<td></td>
<td>(411,263 edges)</td>
</tr>
</tbody>
</table>
Multi-label Node Classification

(a) DBIS (Micro-F1)  
(b) MovieLens (Micro-F1)  
(c) Yelp (Micro-F1)  
(d) DBIS (Macro-F1)  
(e) MovieLens (Macro-F1)  
(f) Yelp (Macro-F1)
### Node Clustering

#### (g) DBIS (NMI)

#### (h) MovieLens (NMI)

#### (i) Yelp (NMI)

#### (j) DBIS (AMI)

#### (k) MovieLens (AMI)

#### (l) Yelp (AMI)
## Link Prediction

AUC with different operators for feature engineering.

<table>
<thead>
<tr>
<th>Method</th>
<th>Operator</th>
<th>DBIS</th>
<th>MOVIE</th>
<th>YELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>Hadamard</td>
<td>0.6367</td>
<td>0.9110</td>
<td>0.7330</td>
</tr>
<tr>
<td></td>
<td>Weighted-L2</td>
<td>0.6094</td>
<td>0.7904</td>
<td>0.6872</td>
</tr>
<tr>
<td>node2vec</td>
<td>Hadamard</td>
<td>0.6362</td>
<td>0.9060</td>
<td>0.6622</td>
</tr>
<tr>
<td></td>
<td>Weighted-L2</td>
<td>0.6292</td>
<td>0.7968</td>
<td>0.6848</td>
</tr>
<tr>
<td>LINE</td>
<td>Hadamard</td>
<td>0.5001</td>
<td>0.8631</td>
<td>0.5689</td>
</tr>
<tr>
<td></td>
<td>Weighted-L2</td>
<td>0.5751</td>
<td>0.7611</td>
<td>0.6229</td>
</tr>
<tr>
<td>HIN2Vec</td>
<td>Hadamard</td>
<td>0.8028</td>
<td>0.9651</td>
<td>0.8117</td>
</tr>
<tr>
<td></td>
<td>Weighted-L2</td>
<td>0.7240</td>
<td>0.7885</td>
<td>0.7137</td>
</tr>
<tr>
<td>metapath2vec</td>
<td>Hadamard</td>
<td>0.6778</td>
<td>0.9151</td>
<td>0.7372</td>
</tr>
<tr>
<td></td>
<td>Weighted-L2</td>
<td>0.7363</td>
<td>0.6996</td>
<td>0.8240</td>
</tr>
<tr>
<td>JUST</td>
<td>Hadamard</td>
<td>0.6463</td>
<td>0.9119</td>
<td>0.7453</td>
</tr>
<tr>
<td></td>
<td>Weighted-L2</td>
<td>0.6260</td>
<td>0.7845</td>
<td>0.6009</td>
</tr>
<tr>
<td>HeGAN</td>
<td>Hadamard</td>
<td>0.9597</td>
<td>0.9207</td>
<td>0.6361</td>
</tr>
<tr>
<td></td>
<td>Weighted-L2</td>
<td>0.6714</td>
<td>0.7970</td>
<td>0.7289</td>
</tr>
<tr>
<td>MARU</td>
<td>Hadamard</td>
<td>0.9979</td>
<td>0.9963</td>
<td>0.7241</td>
</tr>
<tr>
<td></td>
<td>Weighted-L2</td>
<td>0.7468</td>
<td>0.7979</td>
<td>0.8315</td>
</tr>
</tbody>
</table>
Effectiveness of Bidirectional Extended Random Walk

Datasets

- DBIS
- MovieLens

Classical Random Walks
Bidirectional Extended Random Walks

Macro-F1

- 26%
- 28%
- 30%
- 32%
- 34%
- 36%

- 33.32%
- 34.84%
- 26.17%
- 27.9%
Meta-context Space Size vs. Performance

The graph plots the Macro-F1 score against the percentage of labeled nodes for different meta-context space sizes and random walk lengths. The x-axis represents the percentage of labeled nodes (10% to 90%), and the y-axis represents the Macro-F1 score (0.66 to 0.7). The graph shows four lines, each representing a different random walk length: t=1, t=2, t=4, and t=6. The performance generally increases as the percentage of labeled nodes increases and as the random walk length increases.
Conclusions

- Proposed MARU to learn representations for heterogeneous networks.
- Bidirectional extended random walks improves tail node embeddings.
- Meta-context aware skip-gram model dynamically learns the representations without any prior knowledge or manipulation.
- Extensive experiments demonstrate the significant improvements of MARU against state-of-the-art baselines in three practical evaluation tasks with three real-world datasets.
- Two analyses also show the effectiveness of our proposed techniques.

Thank you!
Ask me questions on QA sessions and jyunyu AT cs.ucla.edu
Personal website: https://jyunyu.csie.org/