Heterogeneous Information Learning in Large-Scale Network

Xiao Huang,† Jundong Li,‡ Na Zou,† and Xia (Ben) Hu†

†Texas A&M University, College Station, TX, USA
‡Arizona State University, Tempe, AZ, USA
Emails: {xhuang, nzou1, xiahu}@tamu.edu, jundongl@asu.edu
Roadmap

- Network Embedding
- Heterogeneous Information
- Challenges: Heterogeneity and Large Scale
- Proposed Framework *Heterogeneous Information Learning in Large-Scale Networks* (HILL)
Traditional Network Analysis

Network

Graph Theory

- Shortest path
- Maximum flow
- Graph partition
- Centrality
- ...

Tasks

- Clustering
- Link Prediction
- Classification
- Visualization
- ...

Cut = \{ (S,B), (B,A), (A,C) \}
To take advantage of machine learning, it learns a low-dimenSional vector representation for each node, to preserve the geometrical structure $G$.

Nodes with similar structure $\rightarrow$ similar vectors.

$H$ benefits real-world applications.
Examples of Node Attributes

- Examples: user content in social media, reviews in co-purchasing networks, & paper abstracts in citation networks.
- Rich node attributes are available.
Attributed Networks

- Nodes are not just vertices.
- Node attributes: a rich set of data that describes the unique features of each node.
Heterogeneous Information

- Nodes are accompanied with other types of meaningful information.
  - Node attributes
  - Second-order proximity
  - Link directionality

- Incorporating it into network embedding is potentially helpful in learning better vector representations.
Node Attributes Benefit Embedding

- Node attributes are informative.
- Network and node attributes influence each other and are inherently correlated. (Homophily & social influence)
  - High correlation of user posts and following relationships
  - Strong association between paper topics and citations
Attributes & Network are Correlated

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Scenarios</th>
<th>CorrCoef</th>
<th>Intersect</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlogCatalog</td>
<td>Real-world</td>
<td>3.69e-002</td>
<td>42</td>
<td>0.00e-016</td>
</tr>
<tr>
<td></td>
<td>RandomMean</td>
<td>3.14e-005</td>
<td>7.32</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>RandomMax</td>
<td>1.40e-003</td>
<td>13</td>
<td>4.42e-016</td>
</tr>
<tr>
<td>Flickr</td>
<td>Real-world</td>
<td>1.85e-002</td>
<td>25</td>
<td>0.00e-016</td>
</tr>
<tr>
<td></td>
<td>RandomMean</td>
<td>2.15e-005</td>
<td>3.56</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>RandomMax</td>
<td>5.48e-004</td>
<td>9</td>
<td>3.37e-003</td>
</tr>
</tbody>
</table>

- Hypothesis: there is no correlation between network affinities and node attribute affinities.

- Real-world networks vs randomly-generated networks. Mean and max results on synthetic networks as baselines. A significance level of 0.05
How to Incorporate the Heterogeneous Information?

Heterogeneous Information, e.g., Node Attributes

Embedding Representation

Network Embedding

$H = \begin{bmatrix}
0.54 & 0.27 & n_1 \\
0.22 & 0.91 & n_2 \\
0.55 & 0.28 & n_3 \\
0.98 & 0.11 & n_4 \\
0.32 & 0.87 & n_5 \\
0.26 & 0.11 & n_6 \\
\end{bmatrix}$

$d \ll n$
What if We Have a Large Network?
Real-world Attributed Network are Large

Number of monthly active Facebook users worldwide as of 3rd quarter 2018 (in millions)

Source
Facebook
© Statista 2018

Additional Information:
Worldwide; Facebook; Q3 2008 to Q3 2018
Real-world Node Attributes are High-dimensional

Number of tweets posted by all current MEP per day. (MEP: European Parliament)

The dotted line presents the final day of the latest European Parliament elections

*Calculated on a 31 days rolling average for clarity
Challenges

- Hard to jointly assessing node proximity from heterogeneous information.
  - Node attribute information such as paper abstracts and user posts is distinct from network topological structure
  - Data could be sparse, incomplete and noisy
- Number of nodes and dimension of attributes could be large.
  - Classical algorithms such as eigen-decomposition and gradient descent cannot be applied
  - It could be expensive to store or manipulate the high-dimensional matrices such as node attribute similarity
Heterogeneous Information Learning with Joint Network Embedding

Given $G$ and $A$, we aim to represent each node as a $d$-dimensional row $h_i$, such that $H$ can preserve node proximity both in network and the heterogeneous information.

Examples of $A$: node attributes, second-order proximity, link directionality.
A General Embedding Framework for Heterogeneous Information Learning in Large-Scale Networks, TKDD 2018.

HILL accelerates the optimization by decomposing it into low complexity sub-problems.
Strategies of HILL

1) Assimilate the two info in the similarity space to tackle heterogeneity, but without calculating network similarity matrix.

2) Avoid high-dimensional matrix manipulation.

3) Make sub-problems independent to each other to allow parallel computation.
Strategy 1. Incorporating Node Similarities

Based on the decomposition of attribute similarity and penalty of embedding difference between connected nodes.

\[
\min_H \quad \mathcal{J} = \| S - HH^\top \|_F^2 + \lambda \sum_{(i,j) \in \mathcal{E}} w_{ij} \| h_i - h_j \|_2^2
\]

- $\ell_2$ norm alleviates the impacts from outliers and missing data.
- Fused lasso clusters the network without similarity matrix.
- $\lambda$ adjusts the size of clustering group.
Strategy 2. Avoid High-dimensional Matrix Manipulation

- Make a copy of $\mathbf{H}$ and reformulate into a linearly constrained problem.

\[
\min_{\mathbf{H}} \sum_{i=1}^{n} \| \mathbf{s}_i - \mathbf{h}_i \mathbf{Z}^\top \|_2^2 + \lambda \sum_{(i,j) \in \mathcal{E}} w_{ij} \| \mathbf{h}_i - \mathbf{z}_j \|_2,
\]

subject to \( \mathbf{h}_i = \mathbf{z}_i, \ i = 1, \ldots, n. \)

- Given fixed $\mathbf{H}$, all the row $\mathbf{z}_i$ could be calculated independently.
- Each sub-problem only needs row $\mathbf{s}_i$, not the entire $\mathbf{S}$.
- Time complexity of updating $\mathbf{h}_i$ is $\mathcal{O}(d^3 + dn + d|N(i)|)$, with space complexity $\mathcal{O}(n)$.
- Alternating Direction Method of Multipliers (ADMM) converges to a modest accuracy in a few iterations.
Strategy 3. Enabling Parallel Computation

Worker 1:

Problem 1: $\mathbf{s}_1 = ? \times \mathbf{Z}^T$

Problem 2: $\mathbf{s}_2 = ? \times \mathbf{Z}^T$

Problem 7: $\mathbf{s}_1^T = \mathbf{H} 	imes ?$

Problem 8: $\mathbf{s}_2^T = \mathbf{H} 	imes ?$

Worker 3:

Problem 5: $\mathbf{s}_5 \times ?$

Problem 6: $\mathbf{s}_6 \times ?$

Problem 11: $\mathbf{s}_5^T \times \mathbf{H}$

Problem 12: $\mathbf{s}_6^T \times \mathbf{H}$

Updating

Worker 1:

Problem 1: $\mathbf{s}_1 = \mathbf{Z}^T \times \mathbf{s}_1$

Problem 2: $\mathbf{s}_2 = \mathbf{Z}^T \times \mathbf{s}_2$

Problem 7: $\mathbf{s}_1^T = \mathbf{s}_1 \times \mathbf{H}$

Problem 8: $\mathbf{s}_2^T = \mathbf{s}_2 \times \mathbf{H}$

Worker 3:

Problem 5: $\mathbf{s}_5 \times \mathbf{s}_5$

Problem 6: $\mathbf{s}_6 \times \mathbf{s}_6$

Problem 11: $\mathbf{s}_5^T \times \mathbf{s}_5 	imes \mathbf{H}$

Problem 12: $\mathbf{s}_6^T \times \mathbf{s}_6 	imes \mathbf{H}$
Experimental Settings

- Classification on three real-world network.
  - BlogCatalog (5,196 nodes)
  - Flickr (7,564 nodes)
  - Yelp (249,012 nodes, 1,779,803 edges, 20,000 attribute categories, 47,216,356 entities)

- Three types of baselines.
  - Scalable network embedding: DeepWalk & LINE.
  - Node attribute modeling based on PCA.
  - Attributed network representation learning: MultiSpec & LCMF.

Codes and datasets are available at http://people.tamu.edu/~xhuang/Code.html
HILL outperforms the state-of-the-art embedding algorithms with different latent dimension $d$. 
To answer the second question asked at the beginning of this section, two types of comparisons are shown in Tables 4 and 5.

<table>
<thead>
<tr>
<th>Training Set Percentage</th>
<th>BlogCatalog</th>
<th>Flickr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10%</td>
<td>25%</td>
</tr>
<tr>
<td># nodes for embedding</td>
<td>1,455</td>
<td>2,079</td>
</tr>
</tbody>
</table>

- DeepWalk
- LINE
- HILL_Net
- PCA
- Spectral
- LCMF
- MultiSpec
- HILL_Attri
- HILL_Stream

<table>
<thead>
<tr>
<th>Method</th>
<th>BlogCatalog</th>
<th>Flickr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10%</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>0.491</td>
<td>0.551</td>
</tr>
<tr>
<td></td>
<td>0.433</td>
<td>0.545</td>
</tr>
<tr>
<td></td>
<td>0.556</td>
<td>0.628</td>
</tr>
<tr>
<td></td>
<td>0.695</td>
<td>0.782</td>
</tr>
<tr>
<td></td>
<td>0.717</td>
<td>0.791</td>
</tr>
<tr>
<td></td>
<td>0.778</td>
<td>0.849</td>
</tr>
<tr>
<td></td>
<td>0.678</td>
<td>0.788</td>
</tr>
<tr>
<td></td>
<td>0.841</td>
<td>0.878</td>
</tr>
<tr>
<td></td>
<td>0.770</td>
<td>0.822</td>
</tr>
</tbody>
</table>

- **HILL_Net** uses network only. It employs the second-order proximity of network as the heterogeneous information.
- **HILL_Attri** embeds attributed network.
- For **HILL_Stream**, test nodes come one by one.
Efficiency Evaluation

Fig. 4. Running time of LCMF, MultiSpec and HILL w.r.t. the number of input nodes within a single thread.

Table 6. The Running Time of HILL_A

<table>
<thead>
<tr>
<th></th>
<th>BlogCatalog (sec)</th>
<th>Flickr (sec)</th>
<th>Yelp-sub (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c = 1</td>
<td>26.301</td>
<td>33.751</td>
<td>1065.033</td>
</tr>
<tr>
<td>c = 2</td>
<td>14.233</td>
<td>17.510</td>
<td>581.544</td>
</tr>
</tbody>
</table>

Fig. 5. Impacts of regularization parameter $\lambda$ and penalty parameter $\rho$ on the proposed framework HILL.

- HILL takes much less running time than the attributed network representation learning methods even with single-thread.

7.6 Parameter Analysis

We now answer the third proposed question, i.e., what are the impacts of parameters $\lambda$, $\rho$, and $d$. As discussed in Section 4.3, regularization parameter $\lambda$ in HILL balances the contributions of network information and heterogeneous information proximity. Penalty parameter $\rho$ determines the amount of penalty from the linear constraint $H = Z$. To investigate the impacts of $\lambda$ and $\rho$, we vary $\lambda$ from $10^{-6}$ to $10^{4}$ and $\rho$ from 0.1 to 20. Figure 5 shows the performance in terms of...
Efficiency Evaluation

<table>
<thead>
<tr>
<th></th>
<th>BlogCatalog (sec)</th>
<th>Flickr (sec)</th>
<th>Yelp-sub (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c = 1$</td>
<td>26.301</td>
<td>33.751</td>
<td>1065.033</td>
</tr>
<tr>
<td>$c = 2$</td>
<td>14.233 (-45.9%)</td>
<td>17.510 (-48.1%)</td>
<td>581.544 (-45.4%)</td>
</tr>
</tbody>
</table>

- Running time of HILL w.r.t. the number of workers $c$ on a dual-core processor.
- One of the reasons HILL is efficient: it converges rapidly.
Conclusions

- Nodes are accompanied with other types of meaningful information.
  - Node attributes
  - Second-order proximity
  - Link directionality

- Challenges: Heterogeneity and Large Scale.

- HILL learns low-dimensional vectors to represent all nodes, such that the original network structure and the meaningful heterogeneous information are well preserved.
Acknowledgement

- DATA Lab and collaborators

Data Analytics at Texas A&M (DATA Lab)

- Funding agencies
  - National Science Foundation
  - Defense Advanced Research Projects Agency

- Everyone attending the talk