Accelerated Attributed Network Embedding

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

† Computer Science & Engineering, Texas A&M University, College Station, TX, USA
‡ Computer Science & Engineering, Arizona State University, Tempe, AZ, USA

Emails: {xhuang, xiahu}@tamu.edu, jundongl@asu.edu
What is Network Embedding

Learn a low-dimensional vector representation for each node, such that all the geometrical structure information is preserved.

Similar nodes have similar representations, and the informative latent space benefits real-world applications.
What is Attributed Network

- In real-world information systems, nodes are not just vertices.
- Both node-to-node dependencies & node attribute information are available.
Why Attributes Benefit Embedding

- Node attributes are rich and informative.
- Homophily & social influence: network and node attributes influence each other and are inherently correlated.

• High correlation of user posts and following relationships.
• Strong association between paper topics and citations.
Major Challenges

- Hard to jointly assessing node proximity from heterogeneous information.
  - Node attribute information such as text is distinct from network topological structure.

- Number of nodes and dimension of attributes could be large.
  - Classical algorithms such as eigen-decomposition and gradient descent cannot be applied.
  - It might be expensive to store or manipulate the high-dimensional matrices such as node attribute similarity.
Define Attributed Network Embedding

Given $W$ 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 node attributes.

Nodes with similar topology or attributes would have similar representations.
Major Contributions

- Propose a scalable framework AANE to jointly learn node proximity from network and node attributes.

- Present a distributed optimization algorithm to accelerate by decomposing the task into low complexity sub-problems.

- Strategies for filling the gap:
  
  1. Assimilate the two information in the similarity space to tackle heterogeneity, but without calculating the network similarity matrix.
  
  2. Avoid high-dimensional matrix manipulation.
  
  3. Make sub-problems independent to each other to allow parallel computation.
Based on the decomposition of attribute similarity and penalty of embedding difference between connected nodes.

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

- $\ell_2$ norm alleviates the impacts from outliers and missing data.
- Fused lasso clusters the network without similarity matrix.
- $\lambda$ adjusts the size of clustering group.
Framework AANE: Strategy II

- Make a copy of $H$ and reformulate into a linearly constrained problem.

\[
\min_{H} \sum_{i=1}^{n} \|s_i - h_i Z^T\|_2^2 + \lambda \sum_{(i,j) \in E} w_{ij} \|h_i - z_j\|_2,
\]

subject to $h_i = z_i$, $i = 1, \ldots, n$.

- Given fixed $H$, all the row $z_i$ could be calculated independently.
- Each sub-problem only needs row $s_i$, not the entire $S$.
- Time complexity of updating $h_i$ is $O(d^3 + dn + d|N(i)|)$, with space complexity $O(n)$.
- Alternating Direction Method of Multipliers (ADMM) converges to a modest accuracy in a few iterations.
Framework AANE: Strategy III

Worker 1:

Problem 1: $s_1 = ? \times Z^T$

Problem 2: $s_2 = \hat{q} \times Z^T$

Worker 3:

Problem 5: $s_5 \times Z^T$

Problem 6: $s_6 \times Z^T$

Updating

Problem 7: $s_1^T \times H$

Problem 8: $s_2^T \times H$

Problem 11: $s_5^T \times H$

Problem 12: $s_6^T \times H$
Experimental Setup

- Classification on three real-world network:
  - BlogCatalog
  - Flickr
  - Yelp

- Three types of baselines:
  - Scalable network embedding, DeepWalk & LINE.
  - Node attribute modeling based on PCA.
  - Attributed network representation learning, multispec & LCMF.
AANE achieves higher performance than the state-of-the-art embedding algorithms with different training percentage and latent dimension $d$.  

<table>
<thead>
<tr>
<th>Training Percentage</th>
<th>BlogCatalog</th>
</tr>
</thead>
<tbody>
<tr>
<td># nodes for embedding</td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>20%</td>
<td>2,079</td>
</tr>
<tr>
<td>50%</td>
<td>29,881</td>
</tr>
<tr>
<td>100%</td>
<td>5,196</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>10%</th>
<th>25%</th>
<th>50%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>0.489</td>
<td>0.548</td>
<td>0.606</td>
<td>0.665</td>
</tr>
<tr>
<td>LINE</td>
<td>0.425</td>
<td>0.542</td>
<td>0.620</td>
<td>0.681</td>
</tr>
<tr>
<td>PCA</td>
<td>0.691</td>
<td>0.780</td>
<td>0.821</td>
<td>0.855</td>
</tr>
<tr>
<td>LCMF</td>
<td>0.776</td>
<td>0.847</td>
<td>0.886</td>
<td>0.900</td>
</tr>
<tr>
<td>MultiSpec</td>
<td>0.677</td>
<td>0.787</td>
<td>0.847</td>
<td>0.895</td>
</tr>
<tr>
<td>AANE</td>
<td><strong>0.836</strong></td>
<td><strong>0.875</strong></td>
<td><strong>0.912</strong></td>
<td><strong>0.930</strong></td>
</tr>
</tbody>
</table>

The results show that AANE is statistically significantly better than the two pure network embedding algorithms. As discussed in Section 3.3, two important parameters, 

\[
\begin{align*}
W & = 0.3 \times \text{Micro-average} + 0.7 \times \text{Macro-average} \\
H & = 0.3 \times \text{Micro-average} + 0.7 \times \text{Micro-average}
\end{align*}
\]

are enough to guarantee efficiency and scalability. The proposed mechanism for incorporating attribute information into the embedding representation also raises up. The third contribution of network and node attributes. To jointly improve model, we employ 5-fold cross validation, i.e., randomly divide the entire nodes into a training group (train) and test group have been removed. To investigate the impact of different sizes of training data. The classification performance in terms of percentage is close to 100%, in this subsection. As shown in Figure 3, performance on BlogCatalog is close to 100%.
AANE takes much less running time than the attributed network representation learning methods even with single-thread.

Efficiency Evaluation

- AANE takes much less running time than the attributed network representation learning methods even with single-thread.
The proposed accelerated attributed network embedding (AANE) framework is scalable, efficient, and effective.

Future work:

- Embedding of large-scale and dynamic attributed networks.
- Semi-supervised attributed network embedding.
Acknowledgement

- DATA Lab and collaborators

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

- Everyone attending the talk