TIMME: Twitter Ideology-detection via Multi-task Multi-relational Embedding

Zhiping (Patricia) Xiao, Weiping Song, Haoyan Xu, Zhicheng Ren, Yizhou Sun
KDD’20 Applied Data Science Track
Outline

TIMME
  Motivation
  Contribution
  Data
  Model
  Highlighted Results

Code & Data
Motivation: Ideology Detection

The picture comes from http://www.marekrei.com/blog/political-ideology-detection/.
Motivation: Ideology Detection

https://www.congress.gov/congressional-record
Motivation: Ideology Detection on Twitter

Example: US Presidents on Twitter

Barack Obama
@BarackObama
Dad, husband, President, citizen.
604.2K Following  120.2M Followers
Not followed by anyone you’re following

Donald J. Trump
@realDonaldTrump
45th President of the United States of America
Washington, DC  Instagram.com/realdonaldtrump  Joined March 2009
46 Following  82.5M Followers
Not followed by anyone you’re following

Dan Scavino
@DanScavino

I wrote out some thoughts on how to make this moment a real turning point to bring about real change—and pulled together some resources to help young activists sustain the momentum by channeling their energy into concrete action.
Motivation: Ideology Detection on Twitter

Problem: Ideology Classification on Twitter
Contribution

- TIMME: learning embeddings on sparsely-labeled heterogeneous graph
  - MTL: handles the sparsity of labels
Contribution

- TIMME: learning embeddings on sparsely-labeled heterogeneous graph
  - MTL: handles the sparsity of labels
  - *Optionally* handles incomplete input features

Political-Centered Social Network Dataset

Described in Appendix, released with code
Contribution

- TIMME: learning embeddings on sparsely-labeled heterogeneous graph
  - MTL: handles the sparsity of labels
  - Optionally handles incomplete input features
- Political-Centered Social Network Dataset
  - Described in Appendix, released with code
Data Collection
### Datasets

<table>
<thead>
<tr>
<th></th>
<th>PureP</th>
<th>P50</th>
<th>P20~50</th>
<th>P+all</th>
</tr>
</thead>
<tbody>
<tr>
<td># User</td>
<td>583</td>
<td>5,435</td>
<td>12,103</td>
<td>20,811</td>
</tr>
<tr>
<td># Link</td>
<td>122,347</td>
<td>1,593,721</td>
<td>1,976,985</td>
<td>6,496,107</td>
</tr>
<tr>
<td># Labeled User</td>
<td>581</td>
<td>759</td>
<td>961</td>
<td>1,206</td>
</tr>
<tr>
<td># Featured User</td>
<td>579</td>
<td>5,149</td>
<td>11,725</td>
<td>19,418</td>
</tr>
<tr>
<td># Follow-Link</td>
<td>59,073</td>
<td>529,448</td>
<td>158,746</td>
<td>915,438</td>
</tr>
<tr>
<td># Reply-Link</td>
<td>1,451</td>
<td>96,757</td>
<td>121,133</td>
<td>530,598</td>
</tr>
<tr>
<td># Retweet-Link</td>
<td>19,760</td>
<td>311,359</td>
<td>595,030</td>
<td>1,684,023</td>
</tr>
<tr>
<td># Like-Link</td>
<td>14,381</td>
<td>302,571</td>
<td>562,496</td>
<td>1,794,111</td>
</tr>
<tr>
<td># Mention-Link</td>
<td>27,682</td>
<td>353,586</td>
<td>539,580</td>
<td>1,571,937</td>
</tr>
</tbody>
</table>

Political-Centered Social Network Dataset
Model

Multi-relational GCN

Encoder

|N| Features

Decoder

|N| Embeddings

2|R| Adjacency matrixes
1 Identical matrix

2|R| Tasks

Relation #1
Relation #2
... 
Relation #|R|

Link Prediction

Entity Classification

Layer 1

Layer 2

Multi-relational GCN

loss L

Link Prediction

Entity Classification

Relation #1
Relation #2
... 
Relation #|R|

Tasks
In homogeneous GCN layers:

\[
H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)})
\]

\(\hat{A}\): the normalized adjacency matrix; \(H^{(l+1)}\): layer-\(l\) output; \(H^{(l)}\): layer-\(l\) input; \(W^{(l)}\): layer-\(l\) weight.

In our heterogeneous design:

\[
H^{(l+1)} = \sigma\left(\sum_{r \in \hat{R}} \alpha_r \hat{A}_r H^{(l)} W_r^{(l)}\right)
\]

\(\alpha_r\): attention weight; \(|\hat{R}| = 2R + 1\) includes \(R\) relations, \(R\) reversed relations, 1 identical matrix \(I\).
Model: TIMME Decoder

Entity Classification

Link 1 Prediction

Link 2 Prediction

Link | \( |R| \) Prediction

Fully-Connected Layer

| \( |N| \) Embeddings

Softmax Predictions

Neural Tensor Networks (NTN)

Scores

loss \( L \)

loss \( L_0 \)

loss \( L_1 \)

loss \( L_2 \)

loss \( L_{|R|} \)
Model: TIMME-hierarchical Decoder

Link Prediction

<table>
<thead>
<tr>
<th>Embeddings</th>
<th>Model: TIMME-hierarchical Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>N</td>
</tr>
<tr>
<td>(</td>
<td>N</td>
</tr>
<tr>
<td>( \lambda_1 )</td>
<td>Scores</td>
</tr>
<tr>
<td>(</td>
<td>N</td>
</tr>
<tr>
<td>( \lambda_2 )</td>
<td>Scores</td>
</tr>
<tr>
<td>(</td>
<td>N</td>
</tr>
<tr>
<td>( \lambda_{</td>
<td>R</td>
</tr>
</tbody>
</table>

Link Prediction Tasks

Entity Classification Task

\( |N| \times h \) Output

\( |N| \times 2 \) Output

Softmax

Predictions

Fully-Connected Layer

Neural Tensor Networks (NTN)
TIMME and TIMME-hierarchical:

\[ \mathcal{L} = \sum_{i=0}^{\left| \mathcal{R} \right|} \mathcal{L}_i \]

loss is the sum of losses from all \( |\mathcal{R}| + 1 \) tasks.

- TIMME-hierarchical gives us clues on each relation’s importance to ideology classification via \( \lambda \).

TIMME-single:

\[ \mathcal{L} = \mathcal{L}_i \]

for a single task \( i, i \in \{0, 1, 2, \ldots |\mathcal{R}| \} \).

- TIMME-single proves that multi-task version is better.
### Results Compared with Baselines

<table>
<thead>
<tr>
<th>Model</th>
<th>PureP</th>
<th>P50</th>
<th>P20~50</th>
<th>P=all</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>1.0000/1.0000</td>
<td>0.9600/0.9600</td>
<td>0.9895/0.9895</td>
<td>0.9076/0.9083</td>
</tr>
<tr>
<td>r-GCN</td>
<td>1.0000/1.0000</td>
<td>0.9733/0.9733</td>
<td>0.9895/0.9895</td>
<td>0.9327/0.9333</td>
</tr>
<tr>
<td>HAN</td>
<td>0.9825/0.9824</td>
<td>0.9466/0.9467</td>
<td>0.9789/0.9789</td>
<td>0.9238/0.9250</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>PureP</th>
<th>P50</th>
<th>P20~50</th>
<th>P=all</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMME-single</td>
<td>1.0000/1.0000</td>
<td>0.9733/0.9733</td>
<td>0.9895/0.9895</td>
<td>0.9333/0.9324</td>
</tr>
<tr>
<td>TIMME</td>
<td>0.9825/0.9824</td>
<td>0.9678/0.9678</td>
<td>1.0000/1.0000</td>
<td>0.9495/0.9500</td>
</tr>
<tr>
<td>TIMME-hierarchical</td>
<td>1.0000/1.0000</td>
<td>0.9733/0.9733</td>
<td>0.9895/0.9895</td>
<td>0.9580/0.9583</td>
</tr>
</tbody>
</table>

Table 2: Node classification measured by F1-score/accuracy.

<table>
<thead>
<tr>
<th>Model</th>
<th>PureP</th>
<th>P50</th>
<th>P20~50</th>
<th>P=all</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN+</td>
<td>0.8696/0.8617</td>
<td>0.9593/0.8308</td>
<td>0.9870/0.9576</td>
<td>0.9855/0.9329</td>
</tr>
<tr>
<td>r-GCN</td>
<td>0.8596/0.6091</td>
<td>0.9488/0.8023</td>
<td>0.9872/0.9537</td>
<td>0.9685/0.9201</td>
</tr>
<tr>
<td>HAN+</td>
<td>0.8891/0.7267</td>
<td>0.9598/0.8642</td>
<td>0.9620/0.8850</td>
<td>0.9723/0.9256</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>PureP</th>
<th>P50</th>
<th>P20~50</th>
<th>P=all</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMME-single</td>
<td>0.8809/0.6325</td>
<td>0.9717/0.8792</td>
<td>0.9920/0.9709</td>
<td>0.9936/0.9696</td>
</tr>
<tr>
<td>TIMME</td>
<td>0.8763/0.6324</td>
<td>0.9811/0.9154</td>
<td>0.9945/0.9799</td>
<td>0.9943/0.9736</td>
</tr>
<tr>
<td>TIMME-hierarchical</td>
<td>0.8812/0.6409</td>
<td>0.9899/0.9185</td>
<td>0.9984/0.9813</td>
<td>0.9944/0.9739</td>
</tr>
</tbody>
</table>

Table 3: Link-prediction measured by ROC-AUC/PR-AUC.
County-Level Ideology on Twitter: Florida
Entity-Level Ideology on Twitter: News Agencies

New York Times (@nytimes)
Guardian News (@guardiannews)
CBC News (@cbcnews)
CNN (@CNN)
Christian Science Monitor (@csmonitor)
The American Spectator (@amspectator)
Fox News Opinion (@FoxNewsOpinion)
National Review (@NRO)
Code & Data
Code with data available on Github:

- https://github.com/PatriciaXiao/TIMME

All information included in readme.

Special thanks to: Haoran Wang, Zhiwen Hu, Yupeng Gu