Hyperlink Classification via Structured Graph Embedding

G. Lee, S. Kang, and J. J. Whang*

ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)
July 2019 (*corresponding author)
Real-World Web Graphs

- Hyperlinks are created for different reasons
  - **Navigation links**: navigate the main website
  - **Suggestion links**: suggest users to take a look at related information
  - **Action links**: invoke actions such as ‘edit’, ‘share’, or ‘send an email’

- **Hyperlink Classification Problem**
  - Classify hyperlinks into three classes: navigation, suggestion, and action
Hyperlink Classification

- **Navigation Links**

  ![Google Search for SIGIR 2019](image)
Hyperlink Classification

**Suggestion Links**

- Stack Overflow
- SciPy.org

The `I` attribute only exists on `matrix` objects, not `ndarray`s. You can use `numpy.linalg.inv` to invert arrays:

```python
inverse = numpy.linalg.inv(x)
```

Note that the way you're generating matrices, not all of them will be invertible. You will either need to change the way you're generating matrices, or skip the ones that aren't invertible.

```python
try:
    inverse = numpy.linalg.inv(x)
except numpy.linalg.LinAlgError:
    # Not invertible. Skip this one.
    pass
else:
    # continue with what you were doing
```

SciPy.org documentation:

**numpy.linalg.inv**

Compute the multiplicative inverse of a matrix.

Given a square matrix `A`, return the matrix `Ainv` satisfying `dot(A, Ainv) = dot(Ainv, A) = eye(n)` where `eye(n)` is the `n x n` identity matrix.

Parameters:
- `a`: `ndarray`, `matrix_like`
  - Matrix to be inverted.

Returns:
- `ainv`: `ndarray` or `matrix` (`multiplicative` inverse of the matrix `a`.

Raises:
- LinAlgError
  - If `a` is not square or inversion fails.
Hyperlink Classification

- **Action Links**

![Feature learning from Wikipedia](https://en.wikipedia.org/wiki/Feature_learning)

**Editing Feature learning**

- You are not logged in. Your IP address will be publicly visible if you make any edits. If you log in or create an account, your edits will be attributed to a user name, among other benefits.

Content that violates any copyrights will be deleted. Encyclopedic content must be verifiable. Work submitted to Wikipedia can be edited, used, and redistributed—by anyone—subject to certain terms and conditions.
Real-World Datasets

- Real-world web graphs
  - **Crawling** a set of web pages and hyperlinks starting from a page in Stack Overflow.
  - Conducting a **biased random walk**

|          | $|\mathcal{V}|$ | $|\mathcal{E}|$ | navigation     | suggestion  | action     |
|----------|------------|------------|----------------|-------------|------------|
| web_437  | 404        | 437        | 268 (61.33%)   | 112 (25.63%)| 57 (13.04%)|
| web_1442 | 332        | 1,442      | 1,284 (89.04%) | 93 (6.45%)  | 65 (4.51%) |
| web_10000| 2,202      | 10,000     | 9,892 (98.92%) | 85 (0.85%)  | 23 (0.23%) |

*web_437 and web_1442: some heuristics are applied to balance the class sizes.*
*web_10000 reflects the underlying distribution of the class sizes – very unbalanced.*
Knowledge Graphs

- Graphical Representation of Human Knowledge
  - Each fact is represented by a triplet (**head entity, relation, tail entity**)

**Examples:**
- (Barack Obama, was born in, Honolulu)
- (Barack Obama, is politician of, Honolulu)
- (Barack Obama, is married to, Michelle Obama)
- (Honolulu, located in, United States)
- (Michelle Obama, lives in, United States)
- (Hawaii, has capital, Honolulu)
Knowledge Graph Embedding

- **Representation Learning Technique**
  - Represents entities and relations in a **feature space**.
  - Given a set of **golden triplets** $(S)$ and a set of **corrupted triplets** $(S')$, minimize

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'} \left[ f(h, r, t) + y - f(h', r, t') \right]_+$$

How to compute $f(h, r, t)$ determines different embedding models.
Knowledge Graph Embedding

- **Knowledge Graph Embedding Models**
  - **TransE**: Translating Embeddings for Modeling Multi-relational Data
  - **TransH**: Knowledge Graph Embedding by Translating on Hyperplanes
  - **TransR**: Learning Entity and Relation Embeddings for Knowledge Graph Completion

![Graphical representation of TransE, TransH, and TransR models](image)

Hyperlink Classification Model

- Interpret a **Web Graph** as a **Knowledge Graph**

```
(Barack Obama, was born in, Honolulu)
(Honolulu, located in, United States)
(Michelle Obama, lives in, United States)
...```

```
(NAV, 𝑝₁, 𝑝₃)
(NAV, 𝑝₃, ACTION)
(SUGG, 𝑝₃, 𝑝₂)
...```
Hyperlink Classification Model

- Model Specification and Training
  - A web graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{p_1, p_2, \ldots, p_n\}$, $\mathcal{E} = \{(p_i, p_j) : p_i \in \mathcal{V}, p_j \in \mathcal{V}\}$
  - Each hyperlink has one of the three relation labels $\mathcal{R} = \{n, s, a\}$

$$L = \sum_{(p_i, r, p_j) \in \mathcal{S}} [f(p_i, r, p_j) + \gamma - f(c(p_i, r, p_j))]_+$$

where $c(p_i, r, p_j)$ is defined by

$$c(p_i, r, p_j) = \begin{cases} 
\text{prob. } \alpha/2 : & (p_i, r, q), q \in \mathcal{V}\backslash\{p_j\}, (p_i, r, q) \notin \mathcal{S} \\
\text{prob. } \alpha/2 : & (q, r, p_j), q \in \mathcal{V}\backslash\{p_i\}, (q, r, p_j) \notin \mathcal{S} \\
\text{prob. } (1 - \alpha) : & (p_i, r', p_j), r' \in \mathcal{R}\backslash\{r\} 
\end{cases}$$

$\alpha$ controls the chance to corrupt entities ($0 < \alpha \leq 1$)
Hyperlink Classification Model

- **Prediction**
  - For a directed edge \((p_i, p_j)\) in a test set, the relation label is predicted by
    \[
    r^* = \arg\min_{r \in R} f(p_i, r, p_j)
    \]
  - For TransH embedding model, \(f(p_i, r, p_j)\) is computed by
    \[
    f(p_i, r, p_j) = \|(p_i - w_r^T p_i w_r) + r - (p_j - w_r^T p_j w_r)\|_2^2
    \]
    - \(p_i\) and \(p_j\) : embedding vectors of the pages
    - \(r\) : embedding vector for the relation
    - \(w_r\) : norm vector of the relation-specific hyperplane
### Experimental Results

- F1 scores (%) of our model with different $\alpha$ values and the original TransE, TransH, and TransR.

<table>
<thead>
<tr>
<th></th>
<th>TransE</th>
<th>TransH</th>
<th>TransR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our model, $\alpha = 0.3$</strong></td>
<td>34.29</td>
<td><strong>60.25</strong></td>
<td>57.99</td>
</tr>
<tr>
<td><strong>Our model, $\alpha = 0.5$</strong></td>
<td>34.39</td>
<td>58.87</td>
<td>57.32</td>
</tr>
<tr>
<td><strong>Our model, $\alpha = 0.7$</strong></td>
<td>33.88</td>
<td>58.91</td>
<td><strong>59.83</strong></td>
</tr>
<tr>
<td><strong>The original model</strong></td>
<td><strong>36.22</strong></td>
<td>54.04</td>
<td>53.22</td>
</tr>
</tbody>
</table>

*→ Our model significantly outperforms the original knowledge graph embedding methods.*

<table>
<thead>
<tr>
<th></th>
<th>TransE</th>
<th>TransH</th>
<th>TransR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our model, $\alpha = 0.3$</strong></td>
<td>23.39</td>
<td>53.42</td>
<td><strong>50.04</strong></td>
</tr>
<tr>
<td><strong>Our model, $\alpha = 0.5$</strong></td>
<td><strong>24.86</strong></td>
<td><strong>55.16</strong></td>
<td>46.18</td>
</tr>
<tr>
<td><strong>Our model, $\alpha = 0.7$</strong></td>
<td>21.18</td>
<td>52.70</td>
<td>45.12</td>
</tr>
<tr>
<td><strong>The original model</strong></td>
<td>20.05</td>
<td>29.94</td>
<td>10.35</td>
</tr>
</tbody>
</table>

*→ Creating corrupted triplets by relation perturbation plays a critical role in the hyperlink classification problem.*
## Experimental Results

### F1 score (%) of each class and the average F1 score

<table>
<thead>
<tr>
<th></th>
<th>navigation</th>
<th>suggestion</th>
<th>action</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>web_437 Random-predict</td>
<td>59.75</td>
<td>25.81</td>
<td>11.07</td>
<td>32.21</td>
</tr>
<tr>
<td></td>
<td>60.20</td>
<td>20.96</td>
<td>0.00</td>
<td>27.05</td>
</tr>
<tr>
<td>TransE-original</td>
<td>55.78</td>
<td>31.96</td>
<td>20.93</td>
<td>36.22</td>
</tr>
<tr>
<td>TransH-original</td>
<td>70.80</td>
<td>52.75</td>
<td>38.56</td>
<td>54.04</td>
</tr>
<tr>
<td>TransR-original</td>
<td>67.87</td>
<td>52.86</td>
<td>38.94</td>
<td>53.22</td>
</tr>
<tr>
<td>Our Model</td>
<td><strong>77.04</strong></td>
<td><strong>57.05</strong></td>
<td><strong>46.64</strong></td>
<td><strong>60.25</strong></td>
</tr>
<tr>
<td>web_1442 Random-predict</td>
<td>89.13</td>
<td>5.18</td>
<td>5.65</td>
<td>33.32</td>
</tr>
<tr>
<td></td>
<td>72.98</td>
<td>10.20</td>
<td>36.67</td>
<td>39.95</td>
</tr>
<tr>
<td>TransE-original</td>
<td>42.54</td>
<td>8.57</td>
<td>9.05</td>
<td>20.05</td>
</tr>
<tr>
<td>TransH-original</td>
<td>54.80</td>
<td>13.57</td>
<td>21.45</td>
<td>29.94</td>
</tr>
<tr>
<td>TransR-original</td>
<td>0.00</td>
<td>12.97</td>
<td>18.09</td>
<td>10.35</td>
</tr>
<tr>
<td>Our Model</td>
<td><strong>93.48</strong></td>
<td><strong>22.88</strong></td>
<td><strong>49.12</strong></td>
<td><strong>55.16</strong></td>
</tr>
<tr>
<td>web_10000 Random-predict</td>
<td>98.91</td>
<td>1.60</td>
<td>0.00</td>
<td>33.50</td>
</tr>
<tr>
<td></td>
<td>68.81</td>
<td>1.74</td>
<td>9.92</td>
<td>26.82</td>
</tr>
<tr>
<td>TransE-original</td>
<td>43.25</td>
<td>2.06</td>
<td>0.61</td>
<td>15.31</td>
</tr>
<tr>
<td>TransH-original</td>
<td>63.01</td>
<td>12.02</td>
<td>1.03</td>
<td>25.35</td>
</tr>
<tr>
<td>TransR-original</td>
<td>0.00</td>
<td>5.61</td>
<td>0.61</td>
<td>2.08</td>
</tr>
<tr>
<td>Our Model</td>
<td><strong>99.66</strong></td>
<td><strong>83.22</strong></td>
<td><strong>45.12</strong></td>
<td><strong>76.00</strong></td>
</tr>
</tbody>
</table>

- **Random-predict**: random prediction while preserving the number of hyperlinks in each class.
- **Rule-based**:
  - navigation: within-domain hyperlinks
  - action: ‘edit’, ‘share’, ‘email’, or ‘vote’
  - suggestion: the rest
- **Our model achieves the highest F1 scores.**
Experimental Results

- The average F1, average precision, and average recall
Experimental Results

- Performance on the original web graphs and the randomly shuffled graphs

<table>
<thead>
<tr>
<th></th>
<th>navigation</th>
<th>suggestion</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>web 437</strong></td>
<td>Original</td>
<td>77.04</td>
<td>57.05</td>
</tr>
<tr>
<td></td>
<td>Randomly Shuffled</td>
<td>58.60</td>
<td>25.36</td>
</tr>
<tr>
<td><strong>web 1442</strong></td>
<td>Original</td>
<td>93.48</td>
<td>22.88</td>
</tr>
<tr>
<td></td>
<td>Randomly Shuffled</td>
<td>86.08</td>
<td>6.19</td>
</tr>
<tr>
<td><strong>web 10000</strong></td>
<td>Original</td>
<td>99.66</td>
<td>83.22</td>
</tr>
<tr>
<td></td>
<td>Randomly Shuffled</td>
<td>98.43</td>
<td>1.28</td>
</tr>
</tbody>
</table>

Randomly shuffled graph: the relation labels are randomly shuffled.
Classification performance significantly degrades on the randomly shuffled graphs.
Real-world web graphs have characterized structures in terms of forming each relation type.
→ Enables us to predict the relation labels via structured graph embedding.
Hyperlink Classification in Web Search

- Classify hyperlinks into three classes: navigation, suggestion, and action
- Approach the problem from a structured graph embedding perspective
  - Interpret a web graph as a knowledge graph
  - Modify knowledge graph embedding techniques
- Relation perturbation in negative sampling enables us to significantly improve performance in classifying hyperlinks on web graphs.
More Information: http://bigdata.cs.skku.edu/