Outline

- Introduction: Graphs as Features
- Language Modeling
- DeepWalk
- Evaluation: Network Classification
- Conclusions & Future Work
Features From Graphs

A first step in machine learning for graphs is to extract graph features:

- node: degree
- pairs: # of common neighbors
- groups: cluster assignments

Adjacency Matrix

|V|
What is a Graph Representation?

We can also create features by transforming the graph into a lower dimensional latent representation.

- Anomaly Detection
- Attribute Prediction
- Clustering
- Link Prediction
- ...
DeepWalk learns a latent representation of adjacency matrices using deep learning techniques developed for language modeling.
Visual Example

On Zachary’s Karate Graph:

Input

Output
Advantages of DeepWalk

- Scalable - An online algorithm that does not use entire graph at once
- Walks as sentences metaphor
- Works great!
- Implementation available: bit.ly/deepwalk
Outline

- Introduction: Graphs as Features
- Language Modeling
- DeepWalk
- Evaluation: Network Classification
- Conclusions & Future Work
Language Modeling

Learning a representation means learning a mapping function from word co-occurrence

$$\Phi: v \in V \mapsto \mathbb{R}^{|V| \times d}$$

We hope that the learned representations capture inherent structure

$$||\Phi(\text{rose}) - \Phi(\text{daisy})|| < ||\Phi(\text{rose}) - \Phi(\text{tiger})||$$

We hope that the learned representations capture inherent structure

[Bryan Perozzi, 2009]

[DeepWalk: Online Learning of Social Representations]
This is a very active research topic in NLP.

- **Importance sampling** and **hierarchical classification** were proposed to speed up training.
  
  [F. Morin and Y. Bengio, AISTATS 2005] [Y. Bengio and J. Sencal, IEEENN 2008] [A. Mnih, G. Hinton, NIPS 2008]

- **NLP applications** based on learned representations.
  
  [Colbert et al. *NLP (Almost) from Scratch*, (JMLR), 2011.]

- **Recurrent networks** were proposed to learn sequential representations.
  
  [Tomas Mikolov et al. ICASSP 2011]

- Composed representations learned through **recursive networks** were used for parsing, paraphrase detection, and sentiment analysis.
  

- **Vector spaces** of representations are developed to simplify **compositionality**.
  
  [T. Mikolov, G. Corrado, K. Chen and J. Dean, ICLR 2013, NIPS 2013]
Words frequency in a natural language corpus follows a power law.

- stains open and the moon shining in on the cold, close moon." And neither of the night with the moon shining so bright in the light of the moon. It all boils down under a crescent moon, thrilled by ice the seasons of the moon? Home, alone, dazzling snow, the moon has risen full and the temple of the moon, driving out of
Connection: Power Laws

Vertex frequency in random walks on scale free graphs also follows a power law.
Vertex Frequency in SFG

- Short truncated random walks are sentences in an artificial language!
- Random walk distance is known to be good features for many problems

\[
\begin{align*}
 v_{71} &\rightarrow v_{24} \rightarrow v_5 \rightarrow v_1 \rightarrow v_{17} \rightarrow v_{80} \\
 v_{92} &\rightarrow v_2 \rightarrow v_3 \rightarrow v_1 \rightarrow v_{12} \rightarrow v_{73} \\
 v_{37} &\rightarrow v_{34} \rightarrow v_9 \rightarrow v_1 \rightarrow v_{10} \rightarrow v_{94} \\
 v_{73} &\rightarrow v_{64} \rightarrow v_5 \rightarrow v_1 \rightarrow v_{12} \rightarrow v_{1} \\
 v_{75} &\rightarrow v_{14} \rightarrow v_6 \rightarrow v_1 \rightarrow v_{13} \rightarrow v_{61}
\end{align*}
\]
The Cool Idea

Short random walks = sentences
Outline

- Introduction: Graphs as Features
- Language Modeling
- DeepWalk
- Evaluation: Network Classification
- Conclusions & Future Work
Deep Learning for Networks

1. Input: Graph

2. Random Walks

3. Representation Mapping

4. Hierarchical Softmax

5. Output: Representation

DeepWalk: Online Learning of Social Representations
Deep Learning for Networks

1. Input: Graph

2. Random Walks

3. Representation Mapping

4. Hierarchical Softmax

5. Output: Representation

DeepWalk: Online Learning of Social Representations

Bryan Perozzi
Stony Brook University
Random Walks

- We generate $\gamma$ random walks for each vertex in the graph.

- Each short random walk has length $t$.

- Pick the next step *uniformly* from the vertex neighbors.

- Example:

  $v_{46} \to v_{45} \to v_{71} \to v_{24} \to v_{5} \to v_{1} \to v_{17}$
Deep Learning for Networks

1. Input: Graph

2. Random Walks

3. Representation Mapping

4. Hierarchical Softmax

5. Output: Representation

DeepWalk: Online Learning of Social Representations
Representation Mapping

\[ \mathcal{W}_{v_4} = 4 \]

- Map the vertex under focus \( v_1 \) to its representation.
- Define a window of size \( \mathcal{W} \)
- If \( \mathcal{W} = 1 \) and \( \mathcal{U} = v_1 \)

Maximize:

\[
\Pr(v_3 | \Phi(v_1)) \quad \Pr(v_5 | \Phi(v_1))
\]
Deep Learning for Networks

1. Input: Graph
2. Random Walks
3. Representation Mapping
4. Hierarchical Softmax
5. Output: Representation

\[ \mathcal{W}_{v_4} = \begin{bmatrix} 3 \\ 5 \\ 1 \\ \vdots \end{bmatrix} \]

\[ u_k \rightarrow v_j \rightarrow \Phi \]

\[ \Phi(v_1) \]

DeepWalk: Online Learning of Social Representations
Hierarchical Softmax

Calculating $\Pr(v_3 | \Phi(v_1))$ involves $O(V)$ operations for each update! Instead:

- **Consider the graph vertices as leaves of a balanced binary tree.**
- **Maximizing** $\Pr(v_3 | \Phi(v_1))$ **is equivalent to maximizing the probability of the path from the root to the node.**

Specifically, maximizing

$$\Pr(right | \Phi(v_1); C_2)$$

$$\Pr(left | \Phi(v_1); C_3)$$

$$\Pr(left | \Phi(v_1); C_1)$$

Each of $\{C_1, C_2, C_3\}$ is a logistic binary classifier.
Learning

- Learned parameters:
  - Vertex representations
  - Tree binary classifiers weights

- Randomly initialize the representations.

- For each \{C_1, C_2, C_3\} calculate the loss function.

- Use Stochastic Gradient Descent to update both the classifier weights and the vertex representation \textit{simultaneously}.
Deep Learning for Networks

1. Input: Graph

2. Random Walks

3. Representation Mapping

4. Hierarchical Softmax

5. Output: Representation
Outline

● Introduction: Graphs as Features
● Language Modeling
● DeepWalk
● Evaluation: Network Classification
● Conclusions & Future Work
Attribute Prediction

The Semi-Supervised Network Classification problem:

**INPUT**

A partially labelled graph with node attributes.

**OUTPUT**

Attributes for nodes which do not have them.
Baselines

- Approximate Inference Techniques:
  - weighted vote Relational Neighbor (wvRN) [Macskassy+, ‘03]

- Latent Dimensions
  - Spectral Methods
    - SpectralClustering [Tang+, ‘11]
    - MaxModularity [Tang+, ‘09]
  - k-means
    - EdgeCluster [Tang+, ‘09]
DeepWalk performs well, especially when labels are sparse.

<table>
<thead>
<tr>
<th>Name</th>
<th>BlogCatalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>V</td>
</tr>
<tr>
<td>$</td>
<td>E</td>
</tr>
<tr>
<td>$</td>
<td>Y</td>
</tr>
<tr>
<td>Labels</td>
<td>Interests</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>% Labeled Nodes</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>36.00</td>
<td>38.20</td>
<td>39.60</td>
<td>40.30</td>
<td>41.00</td>
<td>41.30</td>
<td>41.50</td>
<td>41.50</td>
<td>42.00</td>
</tr>
<tr>
<td>SpectralClustering</td>
<td>31.06</td>
<td>34.95</td>
<td>37.27</td>
<td>38.93</td>
<td>39.97</td>
<td>40.99</td>
<td>41.66</td>
<td>42.42</td>
<td>42.62</td>
</tr>
<tr>
<td>EdgeCluster</td>
<td>27.94</td>
<td>30.76</td>
<td>31.85</td>
<td>32.99</td>
<td>34.12</td>
<td>35.00</td>
<td>34.63</td>
<td>35.99</td>
<td>36.29</td>
</tr>
<tr>
<td>Modularity</td>
<td>27.35</td>
<td>30.74</td>
<td>31.77</td>
<td>32.97</td>
<td>34.09</td>
<td>36.13</td>
<td>36.08</td>
<td>37.23</td>
<td>38.18</td>
</tr>
<tr>
<td>wvRN</td>
<td>19.51</td>
<td>24.34</td>
<td>25.62</td>
<td>28.82</td>
<td>30.37</td>
<td>31.81</td>
<td>32.19</td>
<td>33.33</td>
<td>34.28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>BlogCatalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>V</td>
</tr>
<tr>
<td>$</td>
<td>E</td>
</tr>
<tr>
<td>$</td>
<td>Y</td>
</tr>
<tr>
<td>Labels</td>
<td>Interests</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>% Labeled Nodes</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>21.30</td>
<td>23.80</td>
<td>25.30</td>
<td>26.30</td>
<td>27.30</td>
<td>27.60</td>
<td>27.90</td>
<td>28.20</td>
<td>28.90</td>
</tr>
<tr>
<td>SpectralClustering</td>
<td>19.14</td>
<td>23.57</td>
<td>25.97</td>
<td>27.46</td>
<td>28.31</td>
<td>29.46</td>
<td>30.13</td>
<td>31.38</td>
<td>31.78</td>
</tr>
<tr>
<td>EdgeCluster</td>
<td>16.16</td>
<td>19.16</td>
<td>20.48</td>
<td>22.00</td>
<td>23.00</td>
<td>23.64</td>
<td>23.82</td>
<td>24.61</td>
<td>24.92</td>
</tr>
<tr>
<td>Modularity</td>
<td>17.36</td>
<td>20.00</td>
<td>20.80</td>
<td>21.85</td>
<td>22.65</td>
<td>23.41</td>
<td>23.89</td>
<td>24.20</td>
<td>24.97</td>
</tr>
<tr>
<td>wvRN</td>
<td>6.25</td>
<td>10.13</td>
<td>11.64</td>
<td>14.24</td>
<td>15.86</td>
<td>17.18</td>
<td>17.98</td>
<td>18.86</td>
<td>19.57</td>
</tr>
<tr>
<td>Majority</td>
<td>2.52</td>
<td>2.55</td>
<td>2.52</td>
<td>2.58</td>
<td>2.58</td>
<td>2.63</td>
<td>2.61</td>
<td>2.48</td>
<td>2.62</td>
</tr>
</tbody>
</table>

Table 2: Multi-label classification results in BlogCatalog

Bryan Perozzi

Stony Brook University
## Results: Flickr

<table>
<thead>
<tr>
<th>Name</th>
<th>Flickr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>% Labeled Nodes</th>
<th>1%</th>
<th>2%</th>
<th>3%</th>
<th>4%</th>
<th>5%</th>
<th>6%</th>
<th>7%</th>
<th>8%</th>
<th>9%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeepWalk</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SpectralClustering</td>
<td>27.43</td>
<td>30.11</td>
<td>31.63</td>
<td>32.69</td>
<td>33.31</td>
<td>33.95</td>
<td>34.46</td>
<td>34.81</td>
<td>35.14</td>
<td>35.41</td>
</tr>
<tr>
<td>EdgeCluster</td>
<td>25.75</td>
<td>28.53</td>
<td>29.14</td>
<td>30.31</td>
<td>30.85</td>
<td>31.53</td>
<td>31.75</td>
<td>31.76</td>
<td>32.19</td>
<td>32.84</td>
</tr>
<tr>
<td>Modularity</td>
<td>22.75</td>
<td>25.29</td>
<td>27.3</td>
<td>27.6</td>
<td>28.05</td>
<td>29.33</td>
<td>29.43</td>
<td>28.89</td>
<td>29.17</td>
<td>29.2</td>
</tr>
</tbody>
</table>

| **DeepWalk**    |    |    |    |    |    |    |    |    |    |     |
| SpectralClustering | 13.84 | 17.49 | 19.44 | 20.75 | 21.60 | 22.36 | 23.01 | 23.36 | 23.82 | 24.05 |
| EdgeCluster     | 10.52 | 14.10 | 15.91 | 16.72 | 18.01 | 18.54 | 19.54 | 20.18 | 20.78 | 20.85 |
| Modularity      | 10.21 | 13.37 | 15.24 | 15.11 | 16.14 | 16.64 | 17.02 | 17.1 | 17.14 | 17.12 |
| wvRN            | 1.53 | 2.46 | 2.91 | 3.47 | 4.95 | 5.56 | 5.82 | 6.59 | 8.00 | 7.26 |
| Majority        | 0.45 | 0.44 | 0.45 | 0.46 | 0.47 | 0.44 | 0.45 | 0.47 | 0.47 | 0.47 |

Table: Multi-label classification results in Flickr
### Results: YouTube

| Name                  | $|V|$ | $|E|$ | $|Y|$ | Labels | Groups |
|-----------------------|-----|-----|-----|-------|--------|
|                       | 1,138,499 | 2,990,443 | 47 | Groups |

<table>
<thead>
<tr>
<th>% Labeled Nodes</th>
<th>1%</th>
<th>2%</th>
<th>3%</th>
<th>4%</th>
<th>5%</th>
<th>6%</th>
<th>7%</th>
<th>8%</th>
<th>9%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DEEPWALK</strong></td>
<td>37.95</td>
<td>39.28</td>
<td>40.08</td>
<td>40.78</td>
<td>41.32</td>
<td>41.72</td>
<td>42.12</td>
<td>42.48</td>
<td>42.78</td>
<td>43.05</td>
</tr>
<tr>
<td>SpectralClustering</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>EdgeCluster</td>
<td>23.90</td>
<td>31.68</td>
<td>35.53</td>
<td>36.76</td>
<td>37.81</td>
<td>38.63</td>
<td>38.94</td>
<td>39.46</td>
<td>39.92</td>
<td>40.07</td>
</tr>
<tr>
<td>Modularity</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>wvRN</td>
<td>26.79</td>
<td>29.18</td>
<td>33.1</td>
<td>32.88</td>
<td>35.76</td>
<td>37.38</td>
<td>38.21</td>
<td>37.75</td>
<td>38.68</td>
<td>39.42</td>
</tr>
<tr>
<td>Majority</td>
<td>24.90</td>
<td>24.84</td>
<td>25.25</td>
<td>25.23</td>
<td>25.22</td>
<td>25.33</td>
<td>25.31</td>
<td>25.34</td>
<td>25.38</td>
<td>25.38</td>
</tr>
</tbody>
</table>

| **DEEPWALK**     | 29.22 | 31.83 | 33.06 | 33.90 | 34.35 | 34.66 | 34.96 | 35.22 | 35.42 | 35.67 |
| SpectralClustering | —  | —  | —  | —  | —  | —  | —  | —  | —  | —   |
| EdgeCluster      | 19.48 | 25.01 | 28.15 | 29.17 | 29.82 | 30.65 | 30.75 | 31.23 | 31.45 | 31.54 |
| Modularity       | —  | —  | —  | —  | —  | —  | —  | —  | —  | —   |
| wvRN             | 13.15 | 15.78 | 19.66 | 20.9 | 23.31 | 25.43 | 27.08 | 26.48 | 28.33 | 28.89 |

*Spectral Methods do not scale to large graphs.*
Parallelization

- Parallelization doesn’t affect representation quality.

- The sparser the graph, the easier to achieve linear scalability. (Feng+, NIPS ‘11)
Outline

● Introduction: Graphs as Features
● Language Modeling
● DeepWalk
● Evaluation: Network Classification
● Conclusions & Future Work
Variants / Future Work

- Streaming
  - No need to ever store entire graph
  - Can build & update representation as new data comes in.

- "Non-Random" Walks
  - Many graphs occur through as a by-product of interactions
  - One could outside processes (users, etc) to feed the modeling phase
  - [This is what language modeling is doing]
Language Modeling techniques can be used for online learning of network representations.
Thanks!

Bryan Perozzi
@phanein

bperozzi@cs.stonybrook.edu

DeepWalk available at: