LINK PREDICTION USING GRAPH EMBEDDING

Aakash Sinha, Rémy Cazabet
LINK PREDICTION

Time T

Time T+1
LINK PREDICTION

• Numerous applications:
  ‣ Recommender systems (Facebook, Spotify, Amazon,..)
  ‣ Dynamic networks
  ‣ Incomplete datasets
  ‣ …
HEURISTICS

• Historically, link prediction is done using **heuristics**

• Example of heuristics:
  ‣ The more neighbors in common between nodes, the highest chance of having an edge (CN)
  ‣ The highest the degree of nodes, the highest the chance of having an edge (PA)
  ‣ Many others (including more complex ones)
**SUPERVISED LEARNING**

- Heuristics give better results when combined using supervised learning (classifier)

<table>
<thead>
<tr>
<th>Edge</th>
<th>CN</th>
<th>PA</th>
<th>AA</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n1,n2)</td>
<td>0.3</td>
<td>0.2</td>
<td>0.34</td>
<td>Y</td>
</tr>
<tr>
<td>(n1,n3)</td>
<td>0.1</td>
<td>0.34</td>
<td>0.88</td>
<td>N</td>
</tr>
<tr>
<td>(n2,n3)</td>
<td>0.88</td>
<td>0.1</td>
<td>0.55</td>
<td>N</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Edge</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n1,n2)</td>
<td>0.88</td>
</tr>
<tr>
<td>(n1,n3)</td>
<td>0.5</td>
</tr>
<tr>
<td>(n2,n3)</td>
<td>0.2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Classifier

- The table shows the input data for the classifier, including edge labels and corresponding predictions.
SUPERVISED LEARNING

- Heuristics give better results when combined using supervised learning (classifier)
USING EMBEDDINGS

• Embeddings provide a vector **by node**

• Generating one vector **by edge:**
  ‣ Combine vectors of extremities
  ‣ No theoretical arguments on how to combine
  ‣ Best combine function decided empirically (best results)
    - Usually: Hadamar product
### Supervised Learning

#### Example Table

<table>
<thead>
<tr>
<th>Edge</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>...</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n1,n2)</td>
<td>0.3</td>
<td>0.2</td>
<td>0.34</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>(n1,n3)</td>
<td>0.1</td>
<td>0.34</td>
<td>0.88</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>(n2,n3)</td>
<td>0.88</td>
<td>0.1</td>
<td>0.55</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

#### Classifier

<table>
<thead>
<tr>
<th>Edge</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n1,n2)</td>
<td>0.88</td>
</tr>
<tr>
<td>(n1,n3)</td>
<td>0.5</td>
</tr>
<tr>
<td>(n2,n3)</td>
<td>0.2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
• Embedding could also be used with unsupervised learning

• Distance between vectors in the embedding is related to the probability of having an edge between nodes

• => The inverse of the distance between nodes in the embedding is the prediction
OUR QUESTION

• Previous articles have mostly focused on comparing graph embedding techniques between them

• Can we say that graph embeddings are (unambiguously) outperforming heuristics?
  ‣ If yes, by how much?
  ‣ If no, why and how to improve it?
testing set up

- **Methods** (we should add more!)
  - Node2vec
  - VERSE
  - LE
  - HOPE

- **Graphs** (we should add more!)
  - Facebook
  - AstropPH
  - VK
  - CoCit

---

### Heuristics

<table>
<thead>
<tr>
<th>Heuristics</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Neighbors</td>
<td>$</td>
</tr>
<tr>
<td>Adamic Adar</td>
<td>$\sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log</td>
</tr>
<tr>
<td>Preferential attachment</td>
<td>$</td>
</tr>
<tr>
<td>Jaccard Coefficient</td>
<td>$\frac{</td>
</tr>
<tr>
<td>Resource allocation index</td>
<td>$\sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{</td>
</tr>
</tbody>
</table>

### Table 2: Distance function and operators used for each algorithm.

| Name            | |V| | |E| | Density  |
|-----------------|------------|-----|-----|----------------|
| FACEBOOK [12]   | 4k         | 88k | 0.0055 |
| ASTROPH [11]    | 18k        | 198k| 0.00061|
| VK [19]         | 79k        | 2.7M| 0.00043|
• Difficult choice. Link prediction has **high imbalance** between classes (density of real graphs is very low)
  ‣ =>ROC score is independent from class distribution
  ‣ =>AP is not but some authors prefer it (weights to the first few prediction)
  ‣ =>Precision@k is not a single score, but easy to interpret.

• Chosen ones:
  ‣ Average Precision (AP) (with a realistic unbalance)
  ‣ ROC
  ‣ Precision@k
SUPERVISED OR UNSUPERVISED?

=> Supervised is usually more efficient than unsupervised (but not always that much)
WHICH APPROACH IS BEST?

(a) FACEBOOK

ROC Score

=>Only one embedding outperforms heuristics (VERSE)

(b) ASTROPH

ROC Score

=>Only one embedding outperforms heuristics (VERSE)
WHICH APPROACH IS BEST?

Average Precision (with a realistic
No embedding outperform heuristics

Table 4: AP and ROC scores computed separately for pairs
of nodes at distance 2 and at distance 3 and above. For em-
beddings, VERSE and node2vec were used, the highest value
is reported.

6 CONCLUSION AND FUTURE WORKS

In this article, we have shown using a rigorous evaluation frame-
work that graph embedding based methods alone do not improve
over state of the art methods for link prediction in most settings.
By comparing results for di-ferent types of edge creations, those
at distance 2 and those at a higher distance, we found a possible
explanation: heuristics are particularly e-

cient at predicting edge
creation between nodes at distance 2, while graph embeddings,
because they try to capture the structure of the graph at a higher
level, are better at predicting links at a higher distance.

This has important implications for the
field of link prediction: instead of trying to design a single embedding that tries to capture
all aspects of the structure of a graph, it might be more e-
cient to
design several algorithms, specialized in capturing di-ferent aspects
of a graph, and combine them using ensemble methods.

The current work could be extended in two directions: on the
one hand, di-
ferent embeddings could be added to the ensemble
methods, in particular methods with a radically di-

erent approach
that captures structural roles [18]. A versatile approach such as
VERSE might also give interesting results by varying the original
similarity matrix to embed.

On the other hand, it might be interesting to try to better un-
derstand what current link prediction methods are able or not to
capture e-
ciently. In the current article, we have focused on the
distinction between pairs of nodes at distance 2 or more, but it
could be possible to run similar evaluations on other properties,
such as the degrees –or other centrality measures– of nodes, their

WHICH APPROACH IS BEST?

Precision \( \@k \) => No embedding outperform heuristics

Fig. 5: Precision\( \@k \) following a raising celebrity they recently discovered, to strangers connecting become they think they have a common topic of interest, etc.

A link prediction algorithm might have a tendency to predict some types of links more than other, a phenomenon that we call systematic bias. In this article, we focus on biases induced by the network topology. More particularly, we focus on three types of biases:

- Graph distance
- Node degree
- Community structure

To evaluate the biases, we study the evolution of the fraction of new edges verifying a given property. Since predicted edges are ordered, from most probable to less probable, we define the fraction \( \@k \), corresponding to the ratio of pairs of nodes satisfying this property among the \( k \) most likely edges. The dataset is selected as previously explained for the precision \( \@k \).

We define the reference value of fraction \( \@k \) as the value among all edges that do appear in the ground truth (positive examples in the test set). In the scenario of a perfect prediction, the fraction \( \@k \) curve should follow the ground truth scenario until 1000 (corresponding to the real number of observed edges in the test sample), and then move towards the value corresponding to the whole dataset (all pairs of nodes not linked in the training set).

5.1 Graph distance

Edges can appear between nodes that were close or far in the graph in term of graph distance, i.e. length of the shortest path between them. It has been observed that most edges appear between nodes at a short distance, a phenomenon often called triangle closure. It is intuitively known in social network by the saying “friends of my friends are my friends”. Since the number of edges appearing at a distance more than two is usually very low, we consider only two cases:

- Short distance link: the new edge appear between nodes that were previously at distance two in the graph.

FACEBOOK

ASTROPH

Precision \( \@k \) => No embedding outperform heuristics
WHY?

- Why embeddings do not outperform heuristics???
  - (While they are much more advanced)
  - (And most published works seem to show the contrary)

<table>
<thead>
<tr>
<th></th>
<th>Heuristics</th>
<th>Embeddings</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.13378</td>
<td>0.02298</td>
</tr>
<tr>
<td>ROC</td>
<td>0.813</td>
<td>0.618</td>
</tr>
<tr>
<td></td>
<td>(a) Distance 2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Heuristics</th>
<th>Embeddings</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.00219</td>
<td>0.00338</td>
</tr>
<tr>
<td>ROC</td>
<td>0.705</td>
<td>0.794</td>
</tr>
<tr>
<td></td>
<td>(b) Distance &gt; 2</td>
<td></td>
</tr>
</tbody>
</table>
• Long distance link: the new edge appear between nodes that were previously at distance three or more.

Fig. 7 presents the fraction @k of short distance link for the different methods. We can make the following observations:

• In the ground truth, most edges appear between nodes at distance 2, although the value is much lower for VK dataset.

• The Heuristic-based approach is highly biased towards predicting short distance links.

• Most other approaches tend to be biased towards short distance links in the first (most probable) predictions, this value later decreases and the fraction often becomes lower than expected when the expected number of edges (1000) is reached.

5.2 Node degree

Most real networks have heterogeneous degree distributions, that can often be approximated by scale free distributions. In those networks, there is a small fraction of nodes of high degrees that concentrate most of the edges. We define this class of nodes, called Hubs, as the 10% of nodes of highest degrees.

Fraction@k of predictions at distance 2

=> Heuristics favor more the “easy” cases
BIASES

Fig. 7: Ratio@k for the graph distance.

- Long distance link: the new edge appear between nodes that were previously at distance three or more.

Fig. 7 presents the fraction@k of short distance link for the different methods.

We can make the following observations:

- In the ground truth, most edges appear between nodes at distance 2, although the value is much lower for VK dataset.
- The Heuristic-based approach is highly biased towards predicting short distance links.
- Most other approaches tend to be biased towards short distance links in the first (most probable) predictions, this value later decreases and the fraction often becomes lower than expected when the expected number of edges (1000) is reached.

5.2 Node degree

Fig. 9: Fraction@k for High degree nodes (hubs)

Most real networks have heterogeneous degree distributions, that can often be approximated by scale free distributions. In those networks, there is a small fraction of nodes of high degrees that concentrate most of the edges. We define this class of nodes, called Hubs, as the 10% of nodes of highest degrees.

FACEBOOK

ASTROPH

Fraction@k of predictions including Hubs
BIASES

- Possible explanation (positive for embeddings):
  - => Embeddings try to predict “realistic” edges
  - => Heuristics focus only on the “simple” cases, the ones humans think should appear
  - => Heuristics results are more biased, which can be a problem
  - Social networks: recommend only people the most similar to you
  - Product/music recommendation: recommend only the most similar to your previous purchases
  - ...

THANK YOU FOR YOUR ATTENTION

Comments and questions welcomed