Link Prediction Benchmarks

Haifeng Qian
IBM T. J. Watson Research Center
Yorktown Heights, NY
qianhaifeng@us.ibm.com

October 13, 2016

This document describes two temporal link prediction benchmarks that can be downloaded at [10]. Data, code for evaluation and code for several predictors are included in the packages.

Section 1 describes the first benchmark, which is link prediction in coauthorship social networks and is a replication of the experiments in [6]. Performances of several predictors are also presented.

Section 2 describes the second benchmark, which is link prediction in Wikipedia. Performances of several predictors are also presented.

Section 3 describes formats of data files and commands to run and evaluate predictors.

Table 1: Statistics of the coauthorship networks. Entry format is our-number/number-reported-in- [6]. Column Collaborations denotes pairwise relations in the training period. Column $|E_{\text{old}}|$ denotes pairwise relations among Core authors in the training period. Column $|E_{\text{new}}|$ denotes new pairwise relations among Core authors formed in the test period.

<table>
<thead>
<tr>
<th>Training Period</th>
<th>Core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors</td>
<td>Articles</td>
</tr>
<tr>
<td>astro-ph</td>
<td>5321/5343</td>
</tr>
<tr>
<td>cond-mat</td>
<td>5797/5469</td>
</tr>
<tr>
<td>gr-qc</td>
<td>2150/2122</td>
</tr>
<tr>
<td>hep-ph</td>
<td>5482/5414</td>
</tr>
<tr>
<td>hep-th</td>
<td>5295/5241</td>
</tr>
</tbody>
</table>

1 Link prediction in arXiv

The setup in [6] is the following. For five areas in arXiv, given the coauthors of papers published in the training period 1994-1996, the task is to predict new pairwise coauthorship relations formed in the test period 1997-1999. Predictions are only scored for those within Core authors, defined as those who have at least 3 papers in the training period and at least 3 papers in the test period; this Core list is unknown to predictors. Table 1 gives statistics of the five graphs and prediction tasks. Let $E_{\text{new}}$ be the set of new pairwise coauthorship relations among Core authors formed in the test period. Let $E_p$ be the top $|E_{\text{new}}|$ pairs among Core authors that are predicted by a predictor, and the score of this predictor is defined as $|E_p \cap E_{\text{new}}|/|E_{\text{new}}|$.

Table 1 shows that our setup matches [6] closely for four of the five benchmarks. The download package include all five graphs. However, we will focus on benchmarks astro-ph, hep-ph and hep-th, for the following reasons. Benchmark cond-mat differs significantly from that reported in [6], thus is not a valid benchmark to compare against performances reported in [6]. In gr-qc, 131 out of the 397 new relations were formed by a single project which resulted in three papers in the test period, with nearly identical 45-46 authors, [1] being one of the three. Because the size of gr-qc is not large enough relative to this single event, the scores of the predictors are distorted. Thus it is not a surprise that [6] reported that the best predictor for gr-qc is one
Table 2: Comparison of predictor accuracies on coauthorship networks. $A$ denotes the accuracy score of a predictor. $R$ denotes the ratio of $A$ over that of oracle of [6].

<table>
<thead>
<tr>
<th></th>
<th>parameters</th>
<th>astro-ph</th>
<th>hep-ph</th>
<th>hep-th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle of [6]</td>
<td>varying</td>
<td>8.55%</td>
<td>7.20%</td>
<td>7.94%</td>
</tr>
<tr>
<td>PPR, graph #2</td>
<td>$\alpha = 0.50$</td>
<td>8.53%</td>
<td>0.99%</td>
<td>6.74%</td>
</tr>
<tr>
<td>Modified Katz, graph #2</td>
<td>$b_1 = 0.5, b_2 = 0.1, \beta = 0.1, \gamma = 5$</td>
<td>8.41%</td>
<td>0.98%</td>
<td>7.90%</td>
</tr>
<tr>
<td>ERD, graph #1</td>
<td>$b_1 = 0.9, b_2 = 0.9$</td>
<td>8.42%</td>
<td>0.98%</td>
<td>8.16%</td>
</tr>
<tr>
<td>ERD, graph #2</td>
<td>$b_1 = 0.9, b_2 = 0.9, \gamma = 4$</td>
<td>9.55%</td>
<td>1.12%</td>
<td>7.14%</td>
</tr>
</tbody>
</table>

that deletes 85-90% of edges as a preprocessing step, and that the same predictor delivers poor performance on the other benchmarks.

Table 2 lists the accuracy of several predictors. Graph #1 refers to a uniformly weighted graph with edge (paper) weight $b_1$ and node (author) weight $b_2$. Graph #2 refers to the following graph. Each author is modeled as a node, with node weight:

$$b_{\text{author}} = 1 - (1 - b_2)^{(1/\max(1, \log\gamma m_{\text{author}}))}$$

where $m_{\text{author}}$ is the number of coauthors of this author in the training period. Each paper is modeled as a node, and it connects to and from each of its authors with two directed simple edges. An edge from node X to node Y has edge weight:

$$b_{X,Y} = 1 - (1 - b_1)^{(1/\max(1, \log\gamma d_X))}$$

where $d_X$ is the out degree of node X. $\gamma$ in (1)(2) is a tunable parameter and is scanned with other parameters and reported in Table 2. The listed predictors are:

- Oracle of [6] is the highest score reported for each benchmark, by all predictors including meta-approaches.
- The personalized PageRank (PPR) row uses [7, 8] and $\alpha$ is the teleportation probability. Since edge weights (2) are always equal for out-going edges from a same node, changing $b_1$ has no impact on transition probabilities and hence no impact on PPR scores. Since PageRank does not use node weights, we only need to scan parameter $\alpha$ as was done in [6]. The scores are asymmetric. We experimented with max, min, sum and product of two directional scores, and the max gives the best results and is reported in this row.
- The original Katz measure [5] does not use edge or node weights, and we define a modified Katz measure which does:

$$\text{score}_{\text{Katz}}(A, B) = \sum_{l=1}^{\infty} \beta^l \cdot \sum_{\text{length-}l\text{ A-to-B path } i} p_{i,l}$$

where $\beta$ is a constant parameter, and $p_{i,l}$ is the product of edge weights and intermediate node weights for the $i$th path with length $l$. We experimented with max, min, sum and product of A-to-B and B-to-A scores, and the max gives the best results and is reported here.
- Expected Reliable Distance (ERD) is a distance measure from [9]. 10,000 Monte Carlo samples are used for each data point. For graph #2, the shortest-path distance in a Monte Carlo sample is defined as the shorter between the A-to-B path and the B-to-A path if both exist. We will further discuss its performance on hep-ph in Figure 1.

Figure 1 shows detailed comparisons by receiver-operating-characteristic (ROC) curves. The oracle of [6] is omitted because data is not available. Because the number of negative instances (pairs of authors) is far greater than the number of positive instances, the full ROC curves are not visually interesting. Therefore Figure 1 zooms in on the starting portions of curves where the true positive rate is less than 20% and the
false positive rate is less than 1%; these portions are also the most meaningful in applications. Note that ERD #2 performs well on hep-ph for early guesses at around 5% true positive rate, but it degrades quickly after that and becomes the worst of the three by the 20% rate.

2 Link prediction in Wikipedia

This section switches to concept domain: predicting additions of inter-wikipage citations in Wikipedia. The rationale is that citation links reflect Wikipedia contributors’ perception of relation strength between subjects.

We obtain an inter-wikipage-citation graph from [3] which was based on Wikipedia dumps generated in April/May 2014, and another graph from [4] which was based on those in February/March 2015. In both graphs, each node represents a Wikipedia page, and each directed edge from node A to node B represents a citation on page A to page B. Statistics are shown in Table 3. 4,631,780 of the 2014 nodes are mapped to 2015 nodes by exact name matching, and another 87,368 are mapped to 2015 nodes by redirection data from [4] which are pages that have been renamed or merged. The remaining 12,396 of the 2014 nodes cannot be mapped: the majority are Wikipedia pages that have been deleted, and some are due to noise in the data collection of [3, 4]. Such noise is a small fraction and has negligible impact to our measurements.

Table 3: Statistics of Wikipedia citation networks and prediction tasks. \( n_{\text{page}} \) denotes the number of pages; \( d_{2014} \) and \( d_{2015} \) denote the average number of out-going citations on a 2014/2015 page; \( d_{2014, \text{unique}} \) and \( d_{2015, \text{unique}} \) denote the average number of unique out-going citations on a 2014/2015 page; \( n_{\text{prediction}} \) denotes the average number of additions to predict per task.

<table>
<thead>
<tr>
<th></th>
<th>( n_{\text{page}} )</th>
<th>( d_{2014} )</th>
<th>( d_{2014, \text{unique}} )</th>
<th>( d_{2015} )</th>
<th>( d_{2015, \text{unique}} )</th>
<th>( n_{\text{prediction}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014 all pages</td>
<td>4731544</td>
<td>24.66</td>
<td>23.99</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015 all pages</td>
<td>4964985</td>
<td></td>
<td></td>
<td>24.90</td>
<td>24.23</td>
<td></td>
</tr>
<tr>
<td>qualified tasks</td>
<td>93845</td>
<td>156.1</td>
<td>151.0</td>
<td>162.4</td>
<td>157.0</td>
<td>10.06</td>
</tr>
<tr>
<td>training tasks</td>
<td>1000</td>
<td>142.2</td>
<td>137.9</td>
<td>147.5</td>
<td>143.1</td>
<td>10.14</td>
</tr>
<tr>
<td>test tasks</td>
<td>1000</td>
<td>159.0</td>
<td>153.3</td>
<td>165.7</td>
<td>159.6</td>
<td>9.85</td>
</tr>
<tr>
<td>trimmed test tasks</td>
<td>949</td>
<td>157.6</td>
<td>151.7</td>
<td>164.3</td>
<td>157.9</td>
<td>4.63</td>
</tr>
</tbody>
</table>
For each mapped node $A$, we identify $S_{A,2014}$ as the set of 2014 nodes that page $A$ cites in the 2014 graph and that remain in the 2015 graph, $S_{A,2015}$ as the set of 2014 nodes that page $A$ cites in the 2015 graph, and $X_{A,2014}$ as the set of 2014 nodes that cite page $A$. If page $A$ satisfies the condition that $5 \leq |(S_{A,2015} \setminus S_{A,2014}) \setminus X_{A,2014}| \leq |S_{A,2014}| \cdot 20\%$, we consider page $A$ as a qualified prediction task. The rationale behind the size limits is to choose test pages that have undergone thoughtful edits, and their 2014 page contents were already relatively mature; the rationale for excluding in-coming neighbors $X_{A,2014}$ is to make the tasks more challenging, since simple techniques like heavily weighting in-coming edges have no effect. Statistics are shown in Table 3. The number of qualified tasks is large, and we randomly sample a 1000-task training set and a 1000-task test set.

Mean average precision (MAP) [2] is the accuracy score. Nodes that do not exist in the 2015 graph or that exist in $X_{A,2014}$ are removed from a predictor’s outputs before MAP is measured. For each predictor, we choose its parameter(s) ($\alpha$, $\beta$, $b_1$ and $b_2$ to be discussed later) to maximize MAP on the training set, and then under such parameterization, we report its MAP on the test set in Table 4 and Figure 2.

Tasks vary in difficulty. If edits were to make a page more complete, the new links are often reasonably predictable and some are obvious. However, if edits were driven by a recent event, the new links are next to impossible to predict. We form a trimmed test set by removing from the test set targets that are too easy or too difficult. The removed prediction targets for a page $A$ is added to $X_{A,2014}$ so that they are excluded from predictor outputs before MAP evaluation. The results are listed in the last row of Table 3 and the last two columns of Table 4.

Table 4: Comparison of predictor accuracies on additions of inter-wikipage citations in Wikipedia. Each predictor uses its best parameters selected based on the training set. $R$ denotes the ratio of MAP of a predictor over MAP of Adamic/Adar.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>training</th>
<th></th>
<th>test</th>
<th></th>
<th>trimmed test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP</td>
<td>$R$</td>
<td>MAP</td>
<td>$R$</td>
<td>MAP</td>
<td>$R$</td>
</tr>
<tr>
<td>Adamic/Adar</td>
<td>0.0291</td>
<td></td>
<td>0.0281</td>
<td></td>
<td>0.0163</td>
<td></td>
</tr>
<tr>
<td>PPR, graph #1</td>
<td>0.0299</td>
<td>1.029</td>
<td>0.0291</td>
<td>1.038</td>
<td>0.0186</td>
<td>1.140</td>
</tr>
<tr>
<td>PPR, graph #2</td>
<td>0.0321</td>
<td>1.104</td>
<td>0.0309</td>
<td>1.100</td>
<td>0.0206</td>
<td>1.263</td>
</tr>
<tr>
<td>Modified Katz, graph #1</td>
<td>$\beta = 5E-6$</td>
<td></td>
<td>0.0269</td>
<td>0.925</td>
<td>0.0241</td>
<td>0.860</td>
</tr>
<tr>
<td>Modified Katz, graph #2</td>
<td>$b_1 = 0.8$, $b_2 = 0.8$, $\beta = 0.1$, $\gamma = 10$</td>
<td></td>
<td>0.0341</td>
<td>1.173</td>
<td>0.0328</td>
<td>1.170</td>
</tr>
<tr>
<td>ERD, graph #1</td>
<td>$b_1 = 0.4$, $b_2 = 0.9$</td>
<td></td>
<td>0.0266</td>
<td>0.914</td>
<td>0.0233</td>
<td>0.830</td>
</tr>
<tr>
<td>ERD, graph #2</td>
<td>$b_1 = 0.9$, $b_2 = 0.9$, $\gamma = 10$</td>
<td></td>
<td>0.0238</td>
<td>0.817</td>
<td>0.0218</td>
<td>0.778</td>
</tr>
</tbody>
</table>

Table 4 lists accuracy of the various predictors. Graph #1 refers to a uniformly weighted graph with edge weight $b_1$ and node weight $b_2$. Graph #2 refers to a graph with the following weights. For an edge from node $X$ to node $Y$ and that represents the $i$th citation link on page $X$, its weight is

$$b_{\text{edge}} = 1 - (1 - b_1)\frac{\delta_{Y,X}}{\max(1, \log_{\gamma} i) \cdot \max(1, \log_{\gamma} d_{Y,\text{in}})}$$

where

$$\delta_{Y,X} = \begin{cases} 2, & \text{edge exists from } Y \text{ to } X \\ 1, & \text{otherwise} \end{cases}$$

The weight of a node is

$$b_{\text{node}} = 1 - (1 - b_2)^{1/(\log d_{\text{node, in}} + \log d_{\text{node, out}})}$$

The edge weight (4) gives higher weight to a citation link if it is located at an earlier location on a page, or if it points to a less-cited page, or if a returning citation exists. The node weight (6) is a direct adaptation of Adamic/Adar’s (8). For best results, Katz and ERD use (4) as is while PPR uses a linear variant:

$$b_{\text{edge}} = \frac{b_1 \cdot \delta_{Y,X}}{\max(1, \log_{\gamma} i) \cdot \max(1, \log_{\gamma} d_{Y,\text{in}})}$$

Unlike in Section 1, the relations to predict are asymmetric (page $A$ adds a citation to page $B$) and hence there is no requirement for a score definition to be symmetric. Thus all predictors use their one-directional A-to-B score as is:
Adamic/Adar is implemented as:

$$\text{score}_{\text{Adamic/Adar}}(A,B) = \sum_C \frac{n_{A,C} \cdot n_{C,B}}{\log d_{C,\text{in}} + \log d_{C,\text{out}}}$$  \hspace{1cm} (8)$$

where $n_{A,C}$ is the number of A-to-C edges, $n_{C,B}$ is the number of C-to-B edges, and $d_{C,\text{in}}$ and $d_{C,\text{out}}$ are the numbers of in-coming and out-going edges of node C. When multiple node B’s have the same score, we use their in-coming degrees as a tie breaker: a node with higher in-coming degree is ranked first. The same tie breaker is implemented in all other predictors, though its effect is minor to negligible. We use Adamic/Adar accuracy as a reference in Table 4, as it represents what can be achieved through good-quality local analysis.

The personalized PageRank (PPR) rows use [7, 8] and $\alpha$ is the teleportation probability. PPR scores are invariant under different $b_1$ and $b_2$ values and we only need to find the $\alpha$ and $\gamma$ values that maximize its MAP on the training set.

Modified Katz uses (3). A remarkable observation about Modified Katz with graph #2 is that its best performance happens when (3) is almost divergent: with $b_1 = 0.8$, $b_2 = 0.8$ and $\gamma = 10$, the divergence limit for $\beta$ is 0.1075.

Expected Reliable Distance (ERD) is a distance measure from [9]. 100 Monte Carlo samples are used per task. The reduction in the number of samples, compared with Section 1, is because the Wikipedia graph is much larger and denser. Note that [9] used 50 samples in its experiments.

![Figure 2: Accuracy curves on Wikipedia citation network.](image)

Figure 2 shows a more detailed comparison by plotting true positive rate as a function of the number of predictions made. Because the number of negative instances (pages not cited by a page) is far greater than the number of positives, the full curves are not visually interesting, and Figure 2 zooms in on the starting portions.

3 Data formats and command lines

3.1 arXiv data

There are five folders. Each folder is one graph, and contains two files, `kleinberg_star_graph.txt` and `kleinberg_answer.txt`. 
The file `kleinberg_star_graph.txt` describes the topology of graph #2 in Section 1, and one can easily derive other forms of the coauthorship networks. The first line of `kleinberg_star_graph.txt` is:

```
<number of nodes> <number of authors>
```

where the number of nodes is equal to the number authors plus the number of papers. It is followed by `<number of nodes>` lines, each line describing the out-going edges of a node. Nodes have zero-based indexing. The first `<number of authors>` nodes are author nodes, while the rest are paper nodes. Each line has the following format:

```
<out degree> <sink index> <sink index> ⋮ <sink index>
```

The file `kleinberg_answer.txt` gives the correct answers of the prediction task. The first line is

```
<number of core authors>
```

and is followed by `<number of core authors>` lines specifying each of the core authors’ indices. The next line is

```
<number of new pairwise relations>
```

and is followed by `<number of new pairwise relations>` lines, each in the following format:

```
<author one> <author two>
```

where the first index is always the smaller of the two.

There are five `.cpp` files in the download package. The four `predictor_* .cpp` files are source code of the four predictors in Table 2. The `check_answers .cpp` file is source code to check the output of predictors. They each can be compiled into executables simply by

```
g++ -O3 foo .cpp -o foo
```

The command lines to run the four predictors are

```
predictor_pagerank_2 -data kleinberg_star_graph. txt -output myanswers -alpha 0.5
predictor_katz_2 -data kleinberg_star_graph. txt -output myanswers -beta 0.1 -weight1 0.5 -weight2 0.1
predictor_relidist_1 -data kleinberg_star_graph. txt -output myanswers -weight1 0.9 -weight2 0.9
predictor_relidist_2 -data kleinberg_star_graph. txt -output myanswers -weight1 0.9 -weight2 0.9
```

The command line to evaluate the output of a predictor is

```
check_answers -input myanswers -correctanswers kleinberg_answer. txt -report my. report -roc my. roc
```

which produces two text files `my. report` and `my. roc`, the first reporting the accuracy score shown in Table 2 and the second reporting the ROC curve in Figure 1.

### 3.2 Wikipedia data

The file `page_links_en. edges` describes the graph, and files `page_links_en. bidi`, `page_links_en. names` and `page_links_en. gone. index` provide supplemental information. Files `queries. training` and `answers. training` are the 1000 training tasks and their correct answers. Files `queries. test` and `answers. test` are the 1000 test tasks and answers. Files `queries. trimmed. test`, `answers. trimmed. test` and `excluded_answers. trimmed. test` are the trimmed test tasks and answers.

In file `page_links_en. edges`, nodes have zero-based indexing, and each line describes the out-going edges of a node and has the format:

```
<out degree> <sink index> <sink index> ⋮ <sink index>
```
In file `page_links_en.bidi`, each line corresponds to one line in `page_links_en.edges`, and each 0/1 specifies whether each out-going edge has a return citation. Obviously this file is not necessary and can be derived from `page_links_en.edges` on the fly. The purpose here is to simplify predictor code.

The file `page_links_en.gone.index` lists indices of nodes that do not exist in 2015 Wikipedia. Therefore these nodes are removed from output predictions of a predictor before MAP is calculated. This is only used for evaluation purpose and a predictor shall not have access to this information.

The file `page_links_en.names` maps indices to Wikipedia page titles, for your amusement.

Each file `queries.*` lists the starting points of prediction tasks. Each file `answers.*` has the following format per line.

```
<start index> <number of answers> <answer index> <answer index> · · · <answer index>
```

The trimmed test has an extra file `excluded_answers.trimmed.test`, which has the following format per line.

```
<number of entries> <index> <index> · · · <index>
```

These are answers that have been trimmed from the test, and therefore during evaluation these are removed from output predictions of a predictor before MAP is calculated.

There are eight `.cpp` files in the download package. The seven `predictor*.cpp` files are source code of the seven predictors in Table 4. The `check_answers.cpp` file is source code to check the output of predictors. Different from the arXiv data, if we let each predictor write out all its predictions for 1000 tasks, they would take a large amount of disk space. Therefore, we embed evaluation code into the prediction runs by including `check_answers.cpp` in `predictor*.cpp` files. The `predictor*.cpp` files each can be compiled into executables simply by

```
g++ -O3 foo.cpp -o foo
```

With certain gcc versions, extra includes are needed. For example, you may need to add a line `#include <algorithm>` to `check_answers.cpp`.

The command lines to run the seven predictors, and to evaluate their outputs in the same run, are

```
predictor_adamicadar -data page_links_en.edges -query queries.<X> -correctanswers answers.<X> \  
  -ignore page_links_en.gone.index -report my.report -curve my.plot
predictor_pagerank1 -data page_links_en.edges -query queries.<X> -correctanswers answers.<X> \  
  -ignore page_links_en.gone.index -report my.report -curve my.plot \\  
  -alpha 0.5
predictor_pagerank2 -data page_links_en.edges -query queries.<X> -correctanswers answers.<X> \  
  -ignore page_links_en.gone.index -report my.report -curve my.plot \\  
  -bidi page_links_en.bidi -alpha 0.2
predictor_katz1 -data page_links_en.edges -query queries.<X> -correctanswers answers.<X> \  
  -ignore page_links_en.gone.index -report my.report -curve my.plot
predictor_katz2 -data page_links_en.edges -query queries.<X> -correctanswers answers.<X> \  
  -ignore page_links_en.gone.index -report my.report -curve my.plot \\  
  -bidi page_links_en.bidi
predictor_relidist1 -data page_links_en.edges -query queries.<X> -correctanswers answers.<X> \  
  -ignore page_links_en.gone.index -report my.report -curve my.plot
predictor_relidist2 -data page_links_en.edges -query queries.<X> -correctanswers answers.<X> \  
  -ignore page_links_en.gone.index -report my.report -curve my.plot \\  
  -bidi page_links_en.bidi
```

If a run is on the trimmed test set, an extra argument is needed: `-exclude excluded_answers.trimmed.test`.

A run produces two text files `my.report` and `my.plot`, the first reporting the accuracy score shown in Table 4 and the second reporting the curve in Figure 2.
References


