Supervised Rank Aggregation Approach for Link Prediction in Complex Networks

Manisha Pujari & Rushed Kanawati

LIPN - UMR CNRS 7030
Université Paris Nord
99 Av. J.B. Clement 93430, Villetaneuse, FRANCE
manisha.pujari@lipn.univ-paris13.fr

16 April, 2012

Mining Social Network Dynamics Workshop
WWW-2012, Lyon, France
1 Link Prediction

2 Supervised Rank Aggregation based Link Prediction

3 Experiment

4 Conclusion
Problem

Link Prediction

Predicting **new** links between nodes of a graph.

$t_n$ $t_{n+1}$

Applications

- Recommender systems
- Academic/Professional collaborations
- Identification of structures of criminal networks
- Biological networks
Link Prediction Approaches

- **Dyadic**: Computation of link score for unlinked vertices
- **Structural**: Mining rules for evolution of sub-graphs

- **Topology based**: Attributes computed for graph
- **Node-feature based**: Attributes computed for nodes
- **Hybrid**: Combination of the two

- **Temporal**: Consider dynamics of the networks
- **Static**: Do not consider the dynamics of a network
Link Prediction Approaches

- **Dyadic**: Computation of link score for unlinked vertices
- **Topology based**: Attributes computed for graph
- **Temporal**: Consider dynamics of the networks
Dyadic Topological Approaches

Work of [Liben-Nowell & al., 2007]

- Prediction on a co-authorship network.
- For each unlinked node pair \((u, v)\), compute a set of topological attributes \([A_1, A_2, ..., A_n]\).
- Rank all \((u, v)\) based on attribute values.
- Considering only top \(k\) ranked edges as predicted edges, performance of each attribute is found.

Attributes:

- **Neighborhood-based attributes**: Jaccard's coefficient, Common neighbors, Adamic/Adar [Adamic & al. 2003], Preferential attachment etc.
- **Distance-based attributes**: Shortest path distance, Katz [Katz, 1953], Maximum forest algorithm etc.
- **Centrality-based attributes**: PageRank, Degree centrality, Clustering coefficient etc.
Dyadic Topological Approaches

Combining the effect of different topological measures: Application of supervised machine learning algorithms [Benchettara & al., 2010], [Hasan & al., 2006]

Examples:

\[(Nodel_x, Nodel_y) \rightarrow [a_0, a_1, a_2, \ldots, a_n]\]
Dyadic Topological Approaches

Combining the effect of different topological measures: Application of supervised machine learning algorithms
[Benchettara & al., 2010], [Hasan & al., 2006]

Examples:

\[(Node_x, Node_y) \rightarrow [a_0, a_1, a_2, \ldots, a_n]\]

Can we apply rank aggregation methods?
Rank Aggregation (Social choice theory)

⇒ To find an aggregated list with minimum possible disagreement
⇒ Equal weight to all experts

\[ \text{Expert}_1 \implies L_1 = [A, B, C, D] \]
\[ \text{Expert}_2 \implies L_2 = [B, D, A, C] \]
\[ \text{Expert}_3 \implies L_3 = [C, D, A, B] \]
\[ \text{...} \]
\[ \text{...} \]
\[ \text{...} \]

\[ \text{Expert}_n \implies L_n = [D, C, A, B] \]

\[ L_{aggregate} = [?, ?, ?, ?] \]
Distance Measure

- Spearman Footrule Distance: \( F(L_1, L_2) = \sum_{i \in n} |L_1(i) - L_2(i)| \)
- Kendall Tau Distance:
  \[ K(L_1, L_2) = |(i, j) \text{ s.t. } L_1(i) < L_2(j) \& L_1(i) > L_2(j)| \]

Example:

\( L_1 = [A, B, C, D] \) and \( L_2 = [B, D, C, A] \)

\[
F(L_1, L_2) = |L_1(A) - L_2(B)| + |L_1(B) - L_2(B)| + |L_1(C) - L_2(C)| + |L_1(D) - L_2(D)| = 7
\]

\[ K(L_1, L_2) = 4 \]
Borda’s Method [Borda, 1781]

- Based on absolute positioning of elements

\[
B_{L_k}(i) = \{ \text{count}(j) | L_k(j) < L_k(i) \& j \in L_k \}; \quad B(i) = \sum_{t=1}^{k} B_{L_t}(i) \tag{1}
\]

\[
\begin{array}{cccc}
  4 & 3 & 2 & 1 \\
L1 & & & \\
L2 & & & \\
L3 & & & \\
L4 & & & \\
L5 & & & \\
\end{array}
\]

- \(B(L1) = 15\) for L1
- \(B(L2) = 16\) for L2
- \(B(L3) = 9\) for L3
- \(B(L4) = 10\) for L4
Kemeny Optimal Aggregation [Dwork & al., 2001]

Based on relative ranking of elements

\[ SK(\pi, L_1, L_2, L_3, \ldots, L_n) = \sum_{i \in [1, n]} K(\pi, L_i) \]  

\[ (2) \]
Supervised Rank Aggregation

Combining different rankings to get an aggregation giving different weights to the experts

⇒ Proposed approaches

- **Supervised Borda**
- **Supervised local Kemeny**

\[
\begin{align*}
    w_1 &\leftarrow \text{Expert}_1 \implies L_1 \rightarrow [k \text{ elements}] \\
    w_2 &\leftarrow \text{Expert}_2 \implies L_2 \rightarrow [k \text{ elements}] \\
    w_3 &\leftarrow \text{Expert}_3 \implies L_3 \rightarrow [k \text{ elements}] \\
    \vdots & & \vdots \\
    w_n &\leftarrow \text{Expert}_n \implies L_n \rightarrow [k \text{ elements}] 
\end{align*}
\]
Supervised Borda Method

Borda score

\[ B(i) = \sum_{t=1}^{n} w_i \times B_{Lt}(i) \quad \text{where} \quad t \in [1, k] \]  

\[ (3) \]
Supervised Local Kemeny Aggregation

**Steps:**

1. \( L = [L_1, L_2, \ldots, L_n], [w_1, w_2, \ldots, w_n] \), \( m \) elements (\( U \))
2. Initialize \( m \times m \) matrix \( M \) with \( M(x, y) = 0 \)
3. \( \forall (x, y) \in U, \text{ Compute} \)
   \[
   \text{score}(x, y) = \sum_{i=1}^{n} (w_i \times (x \succ y)) \quad \text{where}
   \[
   x \succ y = \begin{cases} 
   0 & \text{if } L_i(x) < L_i(y) \\
   1 & \text{if } L_i(x) > L_i(y)
   \end{cases}
   \]
4. If \( \text{score}(x, y) > 0.5 \times \sum_{i=1}^{n} w_i \), Insert \( M(x, y) = \text{true} \) and \( M(y, x) = \text{false} \)
5. Initial aggregation \( R = L_1 \)
6. For \( x, y \in R \), Swap \( (x, y) \) if \( M(x, y) = \text{false} \)
7. \( R \) is the final aggregation.
Supervised Local Kemeny Aggregation

\[
\begin{array}{c}
\begin{array}{cccc}
  & 4 & 3 & 2 & 1 \\
\hline
0 & L1 & & & \\
1 & L2 & & & \text{false} \\
2 & L3 & & & \text{true} \\
3 & L4 & & & \text{false} \\
4 & L5 & & & \text{true} \\
\end{array}
\end{array}
\]
Examples: \((\text{Node}_x, \text{Node}_y) \rightarrow [a_0, a_1, a_2, \ldots, a_n]\)

Steps:

1. Rank learning examples by attribute values
2. Consider only top \(k\) examples and compute attribute weight \(w_{a_i}\)
3. Rank test examples by attribute to get \(n\) ranked lists
4. Apply supervised rank aggregation
5. Consider only top \(k\) examples of the aggregate list and compute performance.
Computation of attribute weights:

- **Maximization of positive precision:**
  \[ W_{a_i} = n \times \text{Precision}_{a_i} \]  
  (4)

- **Minimization of false positive rate:**
  \[ W_{a_i} = \frac{n}{\text{FPR}_{a_i}} \]  
  (5)
Experiment

- DBLP database

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Training Time</th>
<th>Validation Time</th>
<th>Training examples</th>
<th>Test examples</th>
</tr>
</thead>
</table>

Table: DBLP Datasets

**Performance measure:**

\[ F = \frac{Precision \times Recall}{Precision + Recall} \]  \( (6) \)
Experiment-1: Performance based on learning on complete training dataset

![Graph showing results on complete dataset](image)

- Supervised Borda_prec
- Supervised Borda_fpr
- Naive Bayes
- Decision tree
- KNN
- Borda

**F1-measure**

- Dataset 1
- Dataset 2
- Dataset 3
Experiment-2: Performance in terms of precision based on learning on samples of training dataset

$P$ is the number of positive examples and $N$ is the number of negative examples. In any sample, $N = m \times P$ where $m$ is any positive integer.
Experiment-3: Performance based on learning on complete training dataset and validation on test sample ($N = 5 \times P$).

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Training examples</th>
<th>Test examples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Positive</td>
<td>Total</td>
</tr>
<tr>
<td>Dataset 1</td>
<td>30</td>
<td>1693</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>87</td>
<td>19332</td>
</tr>
<tr>
<td>Dataset 3</td>
<td>102</td>
<td>35190</td>
</tr>
</tbody>
</table>

![Graph](image-url)
Conclusion

- A new definition for supervised rank aggregation.
- Application of supervised rank aggregation to link prediction.

**Future work:**

- Validation on other types of networks like e-commerce networks.
- Application for tag recommendation in folksonomy
  [Pujari & al., 2011],[Pujari & al., 2012]
- Application of our approach with community detection methods.
THANK YOU
References I


References II


