PNA: Partial Network Alignment with Generic Stable Matching

Jiawei Zhang\textsuperscript{1}, Weixiang Shao\textsuperscript{1}, Senzhang Wang\textsuperscript{2}, Xiangnan Kong\textsuperscript{3}, and Philip S. Yu\textsuperscript{1,4}

\textsuperscript{1}University of Illinois at Chicago, Chicago, IL, USA
\textsuperscript{2}Beihang University, Beijing, China
\textsuperscript{3}Worcester Polytechnic Institute, Worcester, MA, USA
\textsuperscript{4}Institute for Data Science, Tsinghua University, Beijing, China
Users participate in multiple social networks simultaneously.
Problem Studied: Social Network Alignment via Shared Common Users
Proposed Network Alignment Framework: PNA (Partial Network Aligner)

- **step 1**: potential anchor link inference with information across networks

**Motivations**: use the heterogeneous information across social networks to infer the existence probabilities of potential anchor links.

- **Motivations**: networks studied in this paper are partially aligned, and each user in a network can be connected to at most one user in another network.

- **step 2**: network matching to prune redundant non-existing anchor links
Step 1: inferring potential anchor links across networks

- Proposed Method: Supervised Anchor Link Prediction

Challenge 1. class imbalance: negative instances >> positive instances

Challenge 2. network heterogeneity: what kind of features can be extracted?

Information used to extract feature vectors for these links:

- Existing anchor links: 
  - \((A_1, A_2)\)
  - \((B_1, B_2)\)
- Non-existing anchor links: 
  - \((A_1, C_2)\)
  - \((E_1, B_2)\)

Link to be predicted: \((F_1, F_2)\)

Supervised learning model

Label/score
**Challenge 1: Class Imbalance**

- **Proposed Solution 1:** **Down Sampling** the Negative Links

  - Distributions of Negative Links in the Feature Space
    - **Safe Negative Links**
    - Borderline Negative Links
    - Noisy Negative Links
    - Redundant Negative Links

  - **training set**
    - **Pos**
    - **Safe Neg**
    - **Borderline Negative**
    - **Noisy Negative**
    - **Redundant Negative**

  - **positive links**
  - **decision boundary**
  - **redundant negative links**
  - **safe negative links**
  - **borderline negative links**
  - **noisy negative links**
Challenge 1: Class Imbalance

- Proposed Solution 2: **Over Sampling** the Positive Links

- Synthetic Positive Links Generation in the Feature Space
  - generate random synthetic positive instances between pairs of positive instances in the feature space
Challenge 2: Network Heterogeneity & Feature Extraction

Information Types: **Who**  **Where**  **What**  **When**

- **Locations**
- **Social Links**
- **Temporal Activities**
  - 8 AM 12 PM 4 PM 8 PM 11 PM
- **Contents: Tweets**
Proposed Solution: Anchor Meta Paths

Bridge node: nodes (besides users) shared across networks

- **Feature extracted for anchor link** \((u^{(i)}, v^{(j)})\) based on anchor meta path \(\Psi\)
- **Number of anchor meta path instances connecting** \(u^{(i)}\) and \(v^{(j)}\)

\[
\text{score}_\Psi(u^{(i)}, v^{(j)}) = |\{\psi | (\psi \in \Psi) \land (u^{(i)} \in T_1) \land (v^{(j)} \in T_k)\}|
\]
Step 2: Network Matching to Prune Non-existing Anchor Links

- Motivations:
  - constraint on anchor links is 1-to-1, according to existing works
  - networks studied in this paper are partially aligned, many users are not connected to anchor links
  - revised constraint on anchor links is 1-to-1≤ (one-to-at most one)
  - how to keep the 1-to-1≤ constraint and prune redundant non-existing anchor links is very challenging
Proposed Solution of 1-to-1 Constraint: Stable Matching

Matching with Block Pair is unstable

Stable Matching
Proposed Solution of $1$-to-$1 \leq$ Constraint: **Self Matching** and **Generic Stable Matching**

- **Self Matching**: users who are shared common users prefer to stay unconnected.
- **Generic Stable Matching**: Stable matching (for shared users) which also allows self matching (unshared users).
- How to do self matching and generic stable matching?
Self Matching and Generic Stable Matching

Preference List

place the user himself at the \((K+1)_{th}\) entry

Truncated Preference List

- K: partial matching rate, used to control the length of users’ preference list, whose sensitivity analysis will be given in the experiments
Pseudo-code of Generic Stable Matching of Networks

Algorithm 1 Generic Gale-Shapley Algorithm

Input: user sets of aligned networks: $U^{(1)}$ and $U^{(2)}$.
- classification results of potential anchor links in $L$
- known anchor links in $A^{(1,2)}$
- truncation rate $K$

Output: a set of inferred anchor links $L'$

1: Initialize the preference lists of users in $U^{(1)}$ and $U^{(2)}$ with predicted existence probabilities of links in $L$ and known anchor links in $A^{(1,2)}$, whose existence probabilities are $1.0$
2: construct the truncated strategies from the preference lists
3: Initialize all users in $U^{(1)}$ and $U^{(2)}$ as free
4: $L' = \emptyset$
5: while $\exists$ free $u^{(1)}_i$ in $U^{(1)}$ and $u^{(1)}_i$’s truncated strategy is non-empty do
6: Remove the top-ranked account $u^{(2)}_j$ from $u^{(1)}_i$’s truncated strategy
7: if $u^{(2)}_j = u^{(1)}_i$ then
8: $L' = L' \cup \{(u^{(1)}_i, u^{(1)}_i)\}$
9: Set $u^{(1)}_i$ as stay unconnected
else
10: if $u^{(2)}_j$ is free then
11: $L' = L' \cup \{(u^{(1)}_i, u^{(2)}_j)\}$
12: Set $u^{(1)}_i$ and $u^{(2)}_j$ as occupied
else
13: $\exists u^{(1)}_p$ that $u^{(2)}_j$ is occupied with.
14: if $u^{(2)}_j$ prefers $u^{(1)}_i$ to $u^{(1)}_p$ then
15: $L' = (L' - \{(u^{(1)}_p, u^{(2)}_j)\}) \cup \{(u^{(1)}_i, u^{(2)}_j)\}$
16: Set $u^{(1)}_p$ as free and $u^{(1)}_i$ as occupied
end if
17: end if
18: end if
19: end while
20: end if
# anchor links: 3,388

Ground truth: existing anchor links

- Hide part of the anchor links, and build models to discover them

## Table I: Properties of the Heterogeneous Networks

<table>
<thead>
<tr>
<th>Property</th>
<th>Twitter</th>
<th>Foursquare</th>
</tr>
</thead>
<tbody>
<tr>
<td># node</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user</td>
<td>5,223</td>
<td>5,392</td>
</tr>
<tr>
<td>tweet/tip</td>
<td>9,490,707</td>
<td>48,756</td>
</tr>
<tr>
<td>location</td>
<td>297,182</td>
<td>38,921</td>
</tr>
<tr>
<td># link</td>
<td></td>
<td></td>
</tr>
<tr>
<td>friend/follow</td>
<td>164,920</td>
<td>76,972</td>
</tr>
<tr>
<td>write</td>
<td>9,490,707</td>
<td>48,756</td>
</tr>
<tr>
<td>locate</td>
<td>615,515</td>
<td>48,756</td>
</tr>
</tbody>
</table>
Experiment Settings

- Comparison Methods:
  - $\text{PNA}_{\text{OMG}}$: Link Prediction (Over Sampling) + Generic Stable Matching
  - $\text{PNA}_{\text{DMG}}$: Link Prediction (Down Sampling) + Generic Stable Matching
  - $\text{PNA}_{\text{OM}}$: Link Prediction (Over Sampling) + Traditional Stable Matching
  - $\text{PNA}_{\text{DM}}$: Link Prediction (Down Sampling) + Traditional Stable Matching
  - $\text{PNA}_O$: Link Prediction (Over Sampling)
  - $\text{PNA}_D$: Link Prediction (Down Sampling)
  - $\text{MNA}$: Link Prediction without Sampling + Traditional Stable Matching
  - $\text{MNA-no}$: Link Prediction without Sampling

<table>
<thead>
<tr>
<th></th>
<th>$\text{PNA}_{\text{OMG}}$</th>
<th>$\text{PNA}_{\text{DMG}}$</th>
<th>$\text{PNA}_{\text{OM}}$</th>
<th>$\text{PNA}_{\text{DM}}$</th>
<th>$\text{PNA}_O$</th>
<th>$\text{PNA}_D$</th>
<th>$\text{MNA}$</th>
<th>$\text{MNA-no}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>over sampling</td>
<td>$\sqrt{}$</td>
<td></td>
<td>$\sqrt{}$</td>
<td></td>
<td>$\sqrt{}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>down sampling</td>
<td></td>
<td>$\sqrt{}$</td>
<td></td>
<td>$\sqrt{}$</td>
<td></td>
<td>$\sqrt{}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>generic stable</td>
<td></td>
<td></td>
<td>$\sqrt{}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>matching</td>
<td></td>
<td></td>
<td></td>
<td>$\sqrt{}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>traditional</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\sqrt{}$</td>
<td></td>
<td></td>
<td>$\sqrt{}$</td>
</tr>
<tr>
<td>stable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>matching</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Evaluation Metrics: Accuracy, AUC, F1
Effectiveness of Sampling Methods

Remarks: PNA\textsubscript{O}, PNA\textsubscript{D} and MNA-no are identical, except
- PNA\textsubscript{O} uses over sampling to handle class imbalance issue
- PNA\textsubscript{D} uses down sampling to deal with class imbalance problem
- MNA-no doesn’t use any sampling methods at all

Observation: PNA\textsubscript{O}, PNA\textsubscript{D} can outperform MNA-no consistently for networks with different \( \eta \) and \( \theta \)

Explanation: Over sampling and down sampling work well in dealing with the class imbalance problem
Experiment Results

\( \eta \): percentage of existing anchor links

### Observations:
1. All the methods achieve better results as \( \eta \) increases
2. Accuracy score achieved by all methods are very high
3. PNAOMG (PNADM) performs better than PNAOM (PNAD)
4. PNAOM and PNADM achieves better results than MNA
5. PNAOM (PNADM and MNA) out-perform PNAO (PNAD and MNA-no)

### Explanations
1. more anchor links, more training instances to build models
2. due to the class imbalance problem, all these methods can make correct prediction of negative links easily and achieve high accuracy
3. generic stable matching and self matching works better for partial network alignment than traditional stable matching
4. over sampling and down sampling works well in addressing the class imbalance problem
5. stable matching is helpful for pruning non-existing anchor links
**Parameter Analysis**

**Observations:** for networks with lower class imbalance rates and alignment rate (e.g., $\theta=5, \eta=0.4$)

- the optimal “partial alignment rate” $K$ for methods $\text{PNA}_{\text{OMG}}$ and $\text{PNA}_{\text{DMG}}$ is 1, i.e., the optimal matching results are candidates with the highest prediction scores
- performance of $\text{PNA}_{\text{OMG}}$ and $\text{PNA}_{\text{DMG}}$ will become worse as $K$ increases from 1 to 5
- as $K$ further increases, it will have no effects on $\text{PNA}_{\text{OMG}}$ and $\text{PNA}_{\text{DMG}}$, as candidates which are far behind in the preference list will never be selected in the matching result

**Observations:** for networks with higher class imbalance rates and alignment rate (e.g., $\theta=50, \eta=0.9$)

- the optimal “partial alignment rate” $K$ for methods $\text{PNA}_{\text{OMG}}$ and $\text{PNA}_{\text{DMG}}$ are 3 and 5 respectively,
- performance of $\text{PNA}_{\text{OMG}}$ ($\text{PNA}_{\text{DMG}}$) will become worse as $K$ increases from 1 to 3 (1 to 5), but drops are $K$ increases to 10
- as $K$ further increases, it will have no effects on $\text{PNA}_{\text{OMG}}$ and $\text{PNA}_{\text{DMG}}$, as candidates which are far behind in the preference list will never be selected in the matching result
Summary

• In this paper, we study the partial network alignment problem.

• A 2-phrase network alignment framework, PNA, is introduced to address the problem
  • step 1: supervised anchor link prediction
    • over sampling/down sampling to handle the class imbalance problem
    • extract features from across the heterogeneous networks based on a set of anchor meta paths
  • step 2: partial network matching with Generic Stable Matching to maintain the $1$-to-$1\leq$ constraint on anchor links
    • self matching is introduced to deal with unshared users
PNA: Partial Network Alignment with Generic Stable Matching

Q&A

Jiawei Zhang¹, Weixiang Shao¹, Senzhang Wang², Xiangnan Kong³, and Philip S. Yu¹,⁴
jzhan9@uic.edu, wshao4@uic.edu, szwang@cse.buaa.edu.cn, xkong@wpi.edu, psyu@cs.uic.edu