Network Alignment: Recent Advances and Future Directions

Si Zhang (UIUC)

Hanghang Tong (UIUC)
Mining on Multiple Networks

Graph Level
(e.g., graph similarity, classification, etc.)

Subgraph Level
(e.g., subgraph matching, cross-domain clustering, etc.)

Node Level
(e.g., network alignment, multi-view node classification)

We Are Here!
Multiple Networks Are Prevalent

Online Social Networks

Transaction Networks
- CHASE
- Bank of America
- PayPal
- Venmo
- Alipay

PPI Networks
- yeast
- elegans
- fly
- mouse

Knowledge Graphs
- DBpedia
- Freebase
- Yago
Multiple Networks: Examples

• Multiple social networks are inter-linked

Linked by “branches”
Multiple Networks: Examples

• Multiple transaction networks are inter-linked

• Q: How to find those “branches”?
What Is Network Alignment?

• Find node correspondence across multiple networks
Network Alignment: Prob. Def.

- **Given:**
  - a set of networks $\{G_l\} \ (l \geq 2)$ where $G_l = \{\mathcal{V}_l, \mathcal{E}_l, A_l\}$;
  - $\mathcal{V}_l, \mathcal{E}_l, A_l$ are the nodes, edges and adjacency matrix of $G_l$;
  - prior alignment matrices $\{H_{l_1, l_2}\}$ between $G_{l_1}$ and $G_{l_2}$.

- **Find:** the alignment matrices $\{S_{l_1, l_2}\}$ between $G_{l_1}$ and $G_{l_2}$. 

![Diagram of networks $G_1$ and $G_2$ with alignment matrices $S_{l_1, l_2}$]
Why Do We Care?

**Identify Species-Specific Pathways**
Protein-Protein Interaction (PPI) networks

PPI network 1  PPI network 2

**Cross Network Information Diffusion**

social network 1  social network 2

**Cross-Site Recommendation**

Users  Items  Users  Items

**Fraud Detection**

looks normal  looks normal  money laundering?
Related Setting: Graph Matching

- It solves for the permutation matrix $P$ that minimizes
\[ \|A_2 - P^T A_1 P\|_F^2 + \text{Tr}(H^T P) \]

- Can be interpreted as a quadratic assignment problem
- $P \in \{0,1\}^{n \times n}$, $P1 = 1$, $1^T P = 1^T$

- Need relaxations on the constraints
  - Doubly stochastic relaxation
  - Spectral relaxation

- Optional external information $H$
Related Setting: Entity Alignment

- To align entities across knowledge graphs

Traditional Methods

• Graph matching-based methods [Koutra’13, Zhang’15]

\[
\min_S \| A_2 - S^T A_1 S \|^2_F
\]

• Assumption: networks are noisy permutations of each other

• Sparse probabilistic relaxation, i.e., \(0 \leq S_{ij} \leq 1\), \(\|S\|_0 \leq t\)

• For bipartite graphs, \(\min_{P,Q} \| B_2 - PB_1 Q \|^2_F\) [Koutra’13]
Traditional Methods

• Random walk-based methods (e.g., IsoRank) [Singh’08, Liao’09]
  • Intuition: random walks on Kronecker product graph
    \[ s = \alpha(A_1 \otimes A_2)s + (1 - \alpha)h \]
  • \( s = \text{vec}(S), \ h = \text{vec}(H) \)

Key Challenge #1: Complexity

• Time complexity:
  • Most of existing works have an at least $O(n^2)$ time complexity
  • Inefficient computations for large-scale networks

• Space complexity:
  • At least $O(n^2)$ to store the alignment matrix
  • Costly memory consumptions

• **Q:** How to efficiently solve network alignment?
Key Challenge #2: Variety

- Networks have rich contextual information
  - Node attributes, e.g., gender, age, etc.
  - Edge attributes, e.g., relation types, etc.

**Q:** How to encode contextual information to enhance the alignment performance?
Key Challenge #3: Heterogeneity

- Networks appear in various sources
  - Networks may capture distinct information
    - Facebook: to connect friend, family, etc.
    - LinkedIn: to connect professionals
  - Same nodes have different behavior patterns
    - E.g., a user is very active in Facebook but quiet in Twitter

**Q:** How to handle the heterogeneity behind multi-sourced networks?
RoadMap

• Motivations and Background

• Part I: Recent Network Alignment Algorithms

• Part II: Network Alignment Applications

• Part III: Future Research Directions
Overview of Part I

Part I: Recent Network Alignment (NA) Algorithms

Pairwise NA
- Consistency-based
  - w/o attributes
  - w/ attributes
- Embedding-based
  - w/o attributes
  - w/ attributes
- Optimal transport-based
  - w/o attributes

Collective NA
- Consistency-based
  - w/o attributes
  - w/ attributes
- Embedding-based
  - w/o attributes

Higher-Order NA
- Consistency-based
  - Single-level
  - Multilevel
- Embedding-based
  - w/o attributes

Related Tasks
- Entity alignment
  - Non-GNN based
  - GNN-based
- Cross-layer inference
- Cross-network transformation
Pairwise Network Alignment

• **Given:** two networks $G_1, G_2$ with/without attributes

• **Find:** the node correspondence across networks

Illustrative example of pairwise network alignment w/o attributes
Consistency-Based Methods

• Intuition:
  • If two nodes are aligned, e.g., node-\(a\) in \(G_1\) and node-\(x\) in \(G_2\)
  • Then their neighbors are likely to be aligned
NetAlign: A Message Passing Method

• Key idea: to maximize the number of overlaps

NetAlign – Formulation #1

- To maximize the # of overlaps
  - Equivalent to maximizing the # of nonzeros in $A$
  - $\frac{\beta}{2} s^T A s$

- $A(i'i', jj') = 1$ if
  - $A_1(i, j) = 1$
  - $A_2(i', j') = 1$
  - $H(i, i') > 0, H(j, j') > 0$

- $s_{ii'}A(i'i', jj')s_{jj'}$ is high if
  - $i, i'$ are likely to be aligned
  - $j, j'$ are likely to be aligned

NetAlign – Formulation #2

• Encode the prior knowledge
  • \( s^T \text{vec}(H) = \sum_{i,i'} S(i, i') H(i, i') \rightarrow \text{score from prior knowledge} \)

• Valid matching constraints
  • \( \sum_{i', \text{s.t. } H(i, i') > 0} S(i, i') \leq 1 \)
  • \( \sum_{i, \text{s.t. } H(i, i') > 0} S(i, i') \leq 1 \)
  • \( S(i, i') \in \{0,1\} \)

NetAlign – Factor Graph

- **Nodes:**
  - Variable nodes: e.g.,
  - Node pairs that form overlaps
  - Function nodes: constraints

\[ f_i = \begin{cases} 
1 & \sum_{H(i,i')>0} s_{ii'} \leq 1 \\
0 & \text{otherwise}
\end{cases} \quad g_i' = \begin{cases} 
1 & \sum_{H(i,i')>0} s_{ii'} \leq 1 \\
0 & \text{otherwise}
\end{cases} \]

\[ h_{ii'jj'} = \begin{cases} 
1 & s_{ii'jj'} = s_{ii'}s_{jj'} \\
0 & \text{otherwise}
\end{cases} \]

- **Edges:** connecting each function node to the variable nodes it acts on

NetAlign – Algorithm

• Belief propagation
  • Iteratively makes local and greedy decisions
  • Updated by passing messages between nodes in factor graph

• Messages $m_{ii'}^t \rightarrow f_i, m_{ii'}^t \rightarrow g_i'$
  • Control matching constraints
  • Also contain info about term $\alpha s^T \text{vec}(H)$

• Messages $m_{ii'}^t \rightarrow h_{ii'jj'}$
  • Agents in a square should communicate
  • Term $\frac{\beta}{2} s^T As$

## Experimental Results

<table>
<thead>
<tr>
<th>Alg.</th>
<th>Data</th>
<th>Overlap</th>
<th>Sol. Time</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MWM</td>
<td>musm-homo</td>
<td>393</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>dmela-scere</td>
<td>135</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>lcsh-small</td>
<td>119</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>lcsh2wiki</td>
<td>2346</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Iso</td>
<td>musm-homo</td>
<td>1027</td>
<td>0.0</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>dmela-scere</td>
<td>301</td>
<td>3.7</td>
<td>10.7</td>
</tr>
<tr>
<td></td>
<td>lcsh-small</td>
<td>257</td>
<td>0.0</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>lcsh2wiki</td>
<td>11732</td>
<td>11.7</td>
<td>587.3</td>
</tr>
<tr>
<td>MP</td>
<td>musm-homo</td>
<td>1076</td>
<td>2.6</td>
<td>13.2</td>
</tr>
<tr>
<td></td>
<td>dmela-scere</td>
<td>369</td>
<td>26.7</td>
<td>34.9</td>
</tr>
<tr>
<td></td>
<td>lcsh-small</td>
<td>316</td>
<td>7.6</td>
<td>12.6</td>
</tr>
<tr>
<td></td>
<td>lcsh2wiki</td>
<td>15974</td>
<td>3795.3</td>
<td>4198.4</td>
</tr>
<tr>
<td>MP++</td>
<td>musm-homo</td>
<td>1062</td>
<td>14.4</td>
<td>17.3</td>
</tr>
<tr>
<td></td>
<td>dmela-scere</td>
<td>376</td>
<td>28.7</td>
<td>33.3</td>
</tr>
<tr>
<td></td>
<td>lcsh-small</td>
<td>318</td>
<td>11.8</td>
<td>15.2</td>
</tr>
<tr>
<td></td>
<td>lcsh2wiki</td>
<td>15771</td>
<td>4103.8</td>
<td>4990.2</td>
</tr>
<tr>
<td>MR</td>
<td>musm-homo</td>
<td>1070</td>
<td>12.5</td>
<td>12.6</td>
</tr>
<tr>
<td></td>
<td>dmela-scere</td>
<td>375</td>
<td>22.7</td>
<td>79.4</td>
</tr>
<tr>
<td></td>
<td>lcsh-small</td>
<td>318</td>
<td>4.1</td>
<td>16.8</td>
</tr>
<tr>
<td></td>
<td>lcsh2wiki</td>
<td>16836</td>
<td>4878.2</td>
<td>4988.0</td>
</tr>
</tbody>
</table>

Final: Attributed Network Alignment

- **Given:**
  - two networks $\{G_l\}$ ($l = 1, 2$) where $G_l = \{\mathcal{V}_l, \mathcal{E}_l, A_l, N_l, E_l\}$
  - $N_l, E_l$ denote the node attributes and edge attributes;
  - prior alignment matrices $H$ between $G_1$ and $G_2$.

- **Find:** the alignment matrix $S$ between $G_1$ and $G_2$.

Final – Formulation #1

• Topological consistency
  • **Intuition:** similar node-pairs tend to have similar neighboring node-pairs

• Example:
  • Large $S(a, x)$
  • Large $A_1(a, b)$ and $A_2(x, y)$  

Final – Formulation #2

- Node attribute consistency
  - **Intuition:** similar node-pairs share similar node attributes

- Large $S(a, x)$ node-$a$ and node-$x$ share similar attributes

---

Final – Formulation #3

• Edge attribute consistency
  • Intuition: similar node-pairs connect to their neighbor-pairs via similar edge attributes

• Example:
  • Large $S(a, x)$
  • Large $S(b, y)$
  → Edge $(a, b)$ & $(x, y)$ share similar attributes

Final – Overall Formulation

- **Objective function**

\[
\min_S J(S) = \sum_{a,b,x,y} \left[ \frac{S(x,a)}{\sqrt{f(x,a)}} - \frac{S(y,b)}{\sqrt{f(y,b)}} \right]^2 \times \Phi(x,a)\Phi(y,b) \times \Psi((x,y),(a,b))
\]

#1. Topology Consistency
#2. Node Attribute Consistency
#3. Edge Attribute Consistency

- **Matrix-form objective function**

\[
\min_S J(S) = \min_S \sum_{v,w} \left[ \frac{s(v)}{\sqrt{D(v,v)}} - \frac{s(w)}{\sqrt{D(w,w)}} \right]^2 W(v,w)
\]

\[s = \text{vec}(S) = \min_s s^T (I - \overline{W}) s\]

attributed Kronecker product

Final – Algorithm

• Fixed-point solution: by setting derivative to 0
  • Converges to the global optimal solution

\[ s = \alpha \mathbf{W} s + (1 - \alpha) \mathbf{h} \Rightarrow s = (1 - \alpha)(I - \alpha \mathbf{W})^{-1} \mathbf{h} \]

• Intuition: a similarity propagation to neighboring node-pairs, which is additionally calibrated by node/edge attributes

• Speed-up variants:
  • Low-rank approximation for full alignment
  • Low-rank approximation for on-query alignment

Final – Low-Rank Approximation Algorithm

• If we only consider node attributes

\[ s = (1 - \alpha) \left( I - \alpha D_N^{-\frac{1}{2}} N (A_1 \otimes A_2) N D_N^{-\frac{1}{2}} \right)^{-1} h \]

• Key Idea: Low rank approximation of \( A_1 \) and \( A_2 \)

\[
\begin{align*}
A_1 & \approx U_1 \Lambda_1 U_1^T \\
A_2 & \approx U_2 \Lambda_2 U_2^T
\end{align*}
\]

\[
\begin{align*}
s & \approx (1 - \alpha) \left( I + \alpha D_N^{-\frac{1}{2}} N U \Lambda U^T N D_N^{-\frac{1}{2}} \right) h \\
\Lambda & = [(\Lambda_1 \otimes \Lambda_2)^{-1} - \alpha U^T N D_N^{-1} N U]^{-1}
\end{align*}
\]

• Complexity: \( O(n^6) \) or \( O(mnt_{\text{max}}) \) \( \rightarrow \) \( O(n^2 r^4) \)

Final – Experimental Results

**Observation**: attributes help improve network alignment.

Final – Experimental Results

Observation: FINAL gains a better quality-speed balance.

Final – Experimental Results

Observation: FINAL On-Query gains around 90% accuracy relative to exact FINAL-N, but more than 100 times faster.

Final – More on Computations

• Further speed-up: from $O(n^2)$ to $O(m)$
  • Key idea: indirect representation of $S$ [1]
  • Theorem: Low-rank of $A_1$ and $A_2 \rightarrow$ low-rank of $S$

```
\begin{align*}
S & \leftarrow U_2 \times M \times U_1^T \\
& \quad r_2 \times r_1
\end{align*}
```

• Alignment quality: linear complexity w/o approximation
  • Multilevel alignment (perfect interpolation theorem) [2]
  • Implicit Krylov subspace methods [3]

Overview of Part I

Part I: Recent Network Alignment (NA) Algorithms

Pairwise NA
- Consistency-based
  - w/o attributes
  - w/ attributes
- Embedding-based
  - w/o attributes
  - w/ attributes
- Optimal transport-based
  - w/o attributes

Collective NA
- Consistency-based
  - w/o attributes
  - w/ attributes
- Embedding-based
  - w/o attributes

Higher-Order NA
- Consistency-based
  - Single-level
  - Multilevel

Related Tasks
- Entity alignment
  - Non-GNN based
  - GNN-based
- Cross-layer inference
- Cross-network transformation
Embedding-Based Methods

• Intuition: to learn node representations that
  • Preserve structural/attribute proximity within networks
  • Preserve proximity across aligned nodes
IONS: Aligning Users by Network Embedding

- **Background:** network embedding by LINE (2nd order)
  - Compute two distributions:

  **Empirical distribution of neighborhood structure:**
  \[
  \hat{p}_2(v_j \mid v_i) = \frac{w_{ij}}{w_{ik}}
  \]

  **Model distribution of neighborhood structure:**
  \[
  p_2(v_j \mid v_i) = \frac{\exp(\bar{u}_i^T \bar{u}_j)}{\sum_{k \in V} \exp(\bar{u}_i^T \bar{u}_k)}
  \]

- **Minimize the KL divergence** by omitting constant terms

  \[
  O_2 = \sum_{i} KL(\hat{p}_2(\cdot \mid v_i), p_2(\cdot \mid v_i)) = \sum_{(i,j) \in E} w_{ij} \log p_2(v_j \mid v_i)
  \]


IONE – Within-Network Embedding

- **Intuition**: to preserve structure proximity

- Embedding vectors for node-\(i\)
  - A node vector \(u_i\)
  - Context vectors: (1) input context \(u_i'\), (2) output context \(u_i''\)

\[
p_1(v_j|v_i) = \frac{\exp (u_j' \cdot u_i)}{\sum_{k=1}^{V} \exp (u_k'^T \cdot u_i)}
\]

\[
p_2(v_i|v_j) = \frac{\exp (u_i'' \cdot u_j)}{\sum_{k=1}^{V} \exp (u_k''^T \cdot u_j)}
\]

**Empirical distributions**: 
\[
\hat{p}_1(i, j) = \frac{w_{ij}}{d_i^{\text{out}}}
\]
\[
\hat{p}_2(i, j) = \frac{w_{ij}}{d_j^{\text{in}}}
\]

- Objective: minimize KL divergences

IONE – Cross-Network Embedding

• **Intuition:** aligned nodes coincide in embedding space

Model distribution: \[ p_1(v_j^Y|v_k^X) = \frac{\exp(u_j^Y^T u_k^X)}{\sum_{k \in V_X} \exp(u_j^Y^T u_k^X)} \]

Empirical distribution: \[ \hat{p}_1(v_j^Y|v_k^X) = \sum_{v_i \in V_Y} p_a(v_i^Y|v_k^X) \times \frac{w_{ij}}{d_{i_{out}}} \]

• \( p_a(v_i^Y|v_k^X) \): probability that \( v_k^X \) and \( v_i^Y \) are aligned

• **Objective:** minimize KL divergences
  • e.g., \( p_1(v_j^Y|v_i^X) \) vs. \( \hat{p}_1(v_j^Y|v_i^X) \)

IONE – Model Inference

• SGD with negative sampling

\[
\begin{align*}
\log p_1(v_j^X | v_i^X) & \propto \log \sigma (\overrightarrow{u_j^X}^T \cdot \overrightarrow{u_i^X}) \\
& \quad + \sum_{m=1}^{K} E_{v_n \sim p_n(v)} \log \sigma (-\overrightarrow{u_n^X}^T \cdot \overrightarrow{u_i^X}) \\
\log p_1(v_j^Y | v_k^X) & \propto \log \sigma (\overrightarrow{u_j^Y}^T \cdot \overrightarrow{u_k^X}) \\
& \quad + \sum_{m=1}^{K} E_{v_n \sim p_n(v)} \log \sigma (-\overrightarrow{u_n^Y}^T \cdot \overrightarrow{u_k^X})
\end{align*}
\]

IONE – Experimental Results

- Dataset: Foursquare-Twitter

IONE – Case Study

DeepLink: Deep Learning for User Identity Linkage

• Motivations:
  • Heterogeneity across networks $\rightarrow$ Complex alignment
  • Scarcity of labeled alignment $\rightarrow$ Supervised training is not easy

• Key questions:
  • How to learn non-linear transformation for alignment?
  • How to boost supervised training algorithm?

• Key idea: use deep neural network with dual-learning

DeepLink – Network Embedding

• Key idea: pre-trained Skip-gram based embedding
  • To predict the context of a center node

• Context sampling:
  • Random walks from center nodes

• Objective function:
  • Original: to maximize
    \[ p(u_{t+j} \mid u_t) = \frac{\exp(v_{u_{t+j}}^T v_{u_t}')}{\sum_{i=1}^m \exp(v_{u_i}^T v_{u_t}')} \]
  • With negative sampling:
    \[ \log[\sigma(v_{u_{t+j}}^T v_{u_t}')] + \sum_{i=1}^K \mathbb{E}_{u_i \sim p_n(u)}[\log(1 - \sigma(v_{u_i}^T v_{u_t}'))] \]

DeepLink – Neural Mapping Learning

• Goal: to learn non-linear alignment across networks

• Intuition: neural networks capture complex nonlinearity

• Key idea: use two multilayer perceptrons as mappings
  • One MLP (denoted by $\Phi$) to map from network $G^s$ to $G^t$
  • Another MLP (denoted by $\Phi^{-1}$) for $G^t$ to $G^s$

DeepLink – Dual Learning

• Goal: to address the lack of labeled alignment

• Components:
  • **Unsupervised alignment pre-training** uses node embedding to learning two weak mapping functions $\Phi$ and $\Phi^{-1}$
  • **Supervised alignment learning** uses labeled alignment to improve weak mapping functions

DeepLink – Unsupervised Pre-training

• Goal: to learn self-consistent mappings
• Method: autoencoder type of architecture
  • Encoder: mapping function $\Phi$
  • Decoder: mapping function $\Phi^{-1}$
• Objective function:
  • Minimize difference between $\Phi^{-1}(\Phi(v_u))$ and $v_u$

DeepLink – Supervised Learning

- Key idea: align according to some reward functions
- Method:
  - Find $k$-similar embeddings $v'(u_i)$ in $G^t$ for mapped embeddings of node-$a$ in $G^s$, i.e., $u_i \in \text{Top}(\Phi(v(u_a)))$
- Rewards:
  - $r_{s,t}^a = \frac{1}{k} \sum_{i=1}^{k} \log(\cos(v(u_i), v'(u_a)) + 1)$
  - $r_{t,s}^a = \frac{1}{k} \sum_{i=1}^{k} \log(\cos(\Phi^{-1}(v'(u_i)), v(u_a)) + 1)$
- To maximize rewards

### DeepLink – Experimental Results

- **Dataset:** Foursquare-Twitter

#### Comparisons of alignment precision.

<table>
<thead>
<tr>
<th></th>
<th>P@1</th>
<th>P@5</th>
<th>P@9</th>
<th>P@13</th>
<th>P@21</th>
<th>P@30</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRW</td>
<td>0.0000</td>
<td>0.0219</td>
<td>0.0476</td>
<td>0.0538</td>
<td>0.0909</td>
<td>0.1603</td>
</tr>
<tr>
<td>MAG</td>
<td>0.0638</td>
<td>0.1362</td>
<td>0.1705</td>
<td>0.2081</td>
<td>0.2708</td>
<td>0.3229</td>
</tr>
<tr>
<td>MAH</td>
<td>0.0500</td>
<td>0.1219</td>
<td>0.1886</td>
<td>0.2148</td>
<td>0.2513</td>
<td>0.0003</td>
</tr>
<tr>
<td>ONE</td>
<td>0.1229</td>
<td>0.2533</td>
<td>0.3038</td>
<td>0.3510</td>
<td>0.4071</td>
<td>0.4270</td>
</tr>
<tr>
<td>IONE</td>
<td>0.2238</td>
<td>0.4033</td>
<td>0.4638</td>
<td>0.5010</td>
<td>0.5571</td>
<td>0.5970</td>
</tr>
<tr>
<td>DeepLink</td>
<td><strong>0.3447</strong></td>
<td><strong>0.5942</strong></td>
<td><strong>0.6609</strong></td>
<td><strong>0.6866</strong></td>
<td><strong>0.7000</strong></td>
<td><strong>0.7048</strong></td>
</tr>
</tbody>
</table>

**Observation:** DeepLink achieves highest accuracy in top-k identity matching.

---

DeepLink – Experimental Results

• Visualization of cosine similarities of randomly sampled anchor nodes (the more diagonalized, the better).

 Observations:
• IONE disrupts the embedding similarities of labeled alignment pairs after training.
• In contrast, DeepLink still preserves the anchor linkage.
• Similarly for testing anchor nodes.

Regal: Representation Learning-Based Graph Alignment

- Goal: unsupervised embedding for network alignment

<table>
<thead>
<tr>
<th></th>
<th>Regal</th>
<th>IONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key idea</td>
<td>matrix factorization</td>
<td>skip-gram w/ negative sampling</td>
</tr>
<tr>
<td>Attributes</td>
<td>w/ node attributes</td>
<td>w/o attributes</td>
</tr>
<tr>
<td>Supervision</td>
<td>unsupervised</td>
<td>semi-supervised</td>
</tr>
<tr>
<td>Complexity</td>
<td>sub-quadratic</td>
<td>sub-quadratic</td>
</tr>
</tbody>
</table>

Regal – Overview

• Node feature extraction
• Node embedding learning by matrix factorization
• Network alignment

Regal – Node Feature Extraction

• Structural identity
  • $\mathcal{R}_u^k$: the set of nodes exactly $k$ steps away from $u$
  • $d_u^k(i)$: the number of nodes in $\mathcal{R}_u^k$ with degree of $i$
  • $d_u = \sum_{i=1}^{K} \delta^{k-1} d_u^k$ ($\delta$ is the discount factor)
  • Logarithmic binning: $d_u^k(i)$ is the number of nodes $u \in \mathcal{R}_u^k$ such that $[\log_2 \deg(u)] = i$

• Attribute-based identity
  • Node input feature vector $f_u$

Regal – Cross-Network Node Similarity

• Direct computation

\[ \text{sim}(u, v) = \exp\left[ -\gamma_s \| \mathbf{d}_u - \mathbf{d}_v \|_2^2 - \gamma_a \times \text{dist}(\mathbf{f}_u, \mathbf{f}_v) \right] \]

• Limitation: costly computation \( O(n^2) \) where \( n = n_1 + n_2 \)

• Efficient computation
  • Reduce to node-landmark similarity
  • \( \mathcal{L} \): a set of \( p \) landmark nodes chosen randomly
  • Node-landmark similarity matrix: \( C(u, v), \ v \in \mathcal{L} \)
  • Landmark-landmark similarity

\[ W(v_1, v_2) = C(v_1, v_2), \ v_1 \in \mathcal{L} \]

Regal – Node Embedding Learning

• Nystrom-based approximation

\[ S \approx \tilde{S} = CW^+ C^T \]

• \( W^+ \): pseudo-inverse of \( W \)

• Embedding: \( Y = CU\Sigma^{1/2} \) where \([U, \Sigma, V] = \text{SVD}(W^+)\)

Regal – Alignment Inference

- K-D tree for fast similarity search
- Similarity scores:

\[ \text{sim}(u, v) = e^{-\|\overline{Y}_1[u] - \overline{Y}_2[v]\|_2^2} \]

- Complexity:
  - Feature extraction: \( O(nKd_{\text{avg}}^2) \)
  - Node similarity: \( O(npb) \)
  - Node embedding: \( O(np^2) \)
  - Alignment: \( O(n \log n) \)

Regal – Experimental Results

- Data constructions: (1) noisy permutations of one network, (2) synthetic node attributes

Regal – Experimental Results

• Running time:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Arxiv (Avg)</th>
<th>PPI (Avg)</th>
<th>Arenas (Avg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINAL</td>
<td>4182 (180)</td>
<td>62.88 (32.20)</td>
<td>3.82 (1.41)</td>
</tr>
<tr>
<td>NetAlign</td>
<td>149.62 (282.03)</td>
<td>22.44 (0.61)</td>
<td>1.89 (0.07)</td>
</tr>
<tr>
<td>IsoRank</td>
<td>17.04 (6.22)</td>
<td>6.14 (1.33)</td>
<td>0.73 (0.05)</td>
</tr>
<tr>
<td>Klaw</td>
<td>1291.00 (373)</td>
<td>476.54 (8.98)</td>
<td>43.04 (0.80)</td>
</tr>
<tr>
<td>REGAL-node2vec</td>
<td>709.04 (20.98)</td>
<td>139.56 (1.54)</td>
<td>15.05 (0.23)</td>
</tr>
<tr>
<td>REGAL-struc2vec</td>
<td>1975.37 (223.22)</td>
<td>441.35 (13.21)</td>
<td>74.07 (0.95)</td>
</tr>
<tr>
<td>REGAL</td>
<td>86.80 (11.23)</td>
<td>18.27 (2.12)</td>
<td>2.32 (0.31)</td>
</tr>
</tbody>
</table>

Faster computations due to landmark strategy and K-D tree search.

Overview of Part I

Part I: Recent Network Alignment (NA) Algorithms

- Pairwise NA
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes
    - w/ attributes
  - Optimal transport-based
    - w/o attributes

- Collective NA
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes

- Higher-Order NA
  - Consistency-based
    - Single-level
    - Multilevel
  - Embedding-based
    - w/o attributes

- Related Tasks
  - Entity alignment
    - Non-GNN based
    - GNN-based
  - Cross-layer inference
  - Cross-network transformation
Gromov-Wasserstein Learning for Graph Matching and Node Embedding

• Backgrounds:
  • Networks are often noisy.
  • Many methods learn specific transformations across embeddings of different networks.

• Key question:
  • How to jointly learn node embeddings and infer alignment?

• Benefits of joint problem:
  • Distance between learned node embeddings as auxiliary information of edges → help reduce noise
  • Learn in same manifold → lower risk of model misspecification

GWL - Preliminaries

• Gromov-Wasserstein distance
  • An optimal transport-like distance for metric spaces
  • Calculates distances between pairs of samples of each domain
  • Measures how these distances compare to those in other domains

• Gromov-Wasserstein discrepancy
  • A relaxation by using dissimilarity measurement instead of strict distance metrics

• Metric-measure space of a graph
  • Corresponds to a pair \((C, \mu) \in \mathbb{R}^{|V| \times |V|} \times \Sigma^{|V|}\) of a graph \(G\).
  • \(C = [c_{ij}]\) represents a node distance/dissimilarity matrix.
  • \(\mu = [\mu_i]\) is the empirical distribution of nodes.

GWL - Gromov-Wasserstein Learning Framework

• Gromov-Wasserstein discrepancy between graphs
  • Given $G_s$ and $G_t$, the discrepancy between $(C_s, \mu_s)$ and $(C_t, \mu_t)$
    
    $$d_{GW}(\mu_s, \mu_t) := \min_{T \in \Pi(\mu_s, \mu_t)} \sum_{i,j,i',j'} L(c^s_{ij}, c^t_{i'j'}) T_{ii'} T_{jj'}$$
    
    $$= \min_{T \in \Pi(\mu_s, \mu_t)} \langle L(C_s, C_t, T), T \rangle.$$  

• $L(\cdot, \cdot)$: element-wise loss, e.g., mean square or KL-divergence

• $T$: optimal transport between nodes of two networks, indicating probabilities of alignment

• $L_{jj'} = \sum_{i,i'} L(c^s_{ij}, c^t_{i'j'}) T_{ii'}$

• $L(C_s, C_t, T) = [L_{jj'}] \in \mathbb{R}^{|V_s| \times |V_t|}$

GWL - Gromov-Wasserstein Learning Framework

• Proposed model
  • Use node embeddings $X_s, X_t$ for dissimilarity matrices

$$\min_{X_s, X_t} \min_{T \in \Pi(\mu_s, \mu_t)} \left\{ \underbrace{L(C_s(X_s), C_t(X_t), T)}_{\text{Gromov-Wasserstein discrepancy}} + \alpha\langle K(X_s, X_t), T \rangle + \beta R(X_s, X_t) \right\}$$

• $C_s(X_s) = (1 - \alpha)C_s + \alpha K(X_s, X_s)$ where $C_s$ is computed by edge weights and $K(X_s, X_s)$ measures distance within same network based on node embedding.

• $R(X_s, X_t) = \sum_{k=S,t} L(K(X_k, X_k), C_k) + L(K(X_s, X_t), C_{st})$

Optional when given labeled alignment

GWL – Learning Algorithm

• Alternatively learn optimal transport and embedding

• Learning optimal transport
  • Proximal point method

\[
\min_{T \in \mathcal{P}(\mu_s, \mu_t)} \langle L(C_s(X_s^{(m)}), C_t(X_t^{(m)}), T), T \rangle + \alpha \langle K(X_s^{(m)}, X_t^{(m)}), T \rangle + \gamma KL(T \| T^{(n)})
\]

A proximal term based on KL-divergence

• Updating embeddings
  • Given optimal transport \( \hat{T}^{(m)} \), solve by gradient descent

\[
\min_{X_s, X_t} \alpha_m \langle K(X_s, X_t), \hat{T}^{(m)} \rangle + \beta R(X_s, X_t)
\]

GWL – Experimental Results

• Communication network alignment
  • Dataset: MC3 used in the Mini-Challenge 3 of VAST Challenge 2018

• Model Variants:
  • GWL-C and GWL-R: use cosine and RBF distance on embeddings
  • GWD: no embedding-based distance

<table>
<thead>
<tr>
<th>Method</th>
<th>Call→Email (Sparse) Node Correctness (%)</th>
<th>Call→Email (Dense) Node Correctness (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAA</td>
<td>34.22</td>
<td>0.53</td>
</tr>
<tr>
<td>LRSA</td>
<td>38.20</td>
<td>2.93</td>
</tr>
<tr>
<td>TAME</td>
<td>37.39</td>
<td>2.67</td>
</tr>
<tr>
<td>GRAAL</td>
<td>39.67</td>
<td>0.48</td>
</tr>
<tr>
<td>MI-GRAAL</td>
<td>35.53</td>
<td>0.64</td>
</tr>
<tr>
<td>MAGNA++</td>
<td>7.88</td>
<td>0.09</td>
</tr>
<tr>
<td>HugAlign</td>
<td>36.21</td>
<td>3.86</td>
</tr>
<tr>
<td>NETAL</td>
<td>36.87</td>
<td>1.77</td>
</tr>
<tr>
<td><strong>GWD</strong></td>
<td><strong>23.16±0.46</strong></td>
<td><strong>1.77±0.22</strong></td>
</tr>
<tr>
<td><strong>GWL-R</strong></td>
<td><strong>39.64±0.57</strong></td>
<td><strong>3.80±0.23</strong></td>
</tr>
<tr>
<td><strong>GWL-C</strong></td>
<td><strong>40.45±0.53</strong></td>
<td><strong>4.23±0.27</strong></td>
</tr>
</tbody>
</table>

GWL – Experimental Results

• Procedure recommendation
  • Dataset: MIMIC-III dataset
  • Goal: recommend suitable procedures for patients, according to their disease characteristics.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1 (%)</th>
<th></th>
<th></th>
<th>Top-5 (%)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
<td>P</td>
<td>R</td>
<td>F1</td>
</tr>
<tr>
<td>Word2Vec</td>
<td>39.95</td>
<td>13.27</td>
<td>18.25</td>
<td>28.89</td>
<td>46.98</td>
<td>32.59</td>
</tr>
<tr>
<td>GloVe</td>
<td>32.66</td>
<td>13.01</td>
<td>17.22</td>
<td>27.93</td>
<td>44.79</td>
<td>31.47</td>
</tr>
<tr>
<td>DWL (Scratch)</td>
<td>37.89</td>
<td>12.42</td>
<td>17.16</td>
<td>27.39</td>
<td>43.81</td>
<td>30.81</td>
</tr>
<tr>
<td>DWL (Finetune)</td>
<td>40.00</td>
<td>13.76</td>
<td>18.71</td>
<td>30.59</td>
<td>48.56</td>
<td>34.28</td>
</tr>
<tr>
<td>GWD-R</td>
<td>46.29</td>
<td>17.01</td>
<td>22.32</td>
<td>31.82</td>
<td>43.81</td>
<td>33.77</td>
</tr>
<tr>
<td>GWD-C</td>
<td>43.16</td>
<td>15.79</td>
<td>20.77</td>
<td>31.42</td>
<td>42.99</td>
<td>33.25</td>
</tr>
<tr>
<td>GWL-R</td>
<td>46.20</td>
<td>16.93</td>
<td>22.22</td>
<td>32.03</td>
<td>44.75</td>
<td>34.18</td>
</tr>
<tr>
<td>GWL-C</td>
<td>47.46</td>
<td>17.25</td>
<td>22.71</td>
<td>32.09</td>
<td>45.64</td>
<td>34.31</td>
</tr>
</tbody>
</table>
Overview of Part I

Part I: Recent Network Alignment (NA) Algorithms

- **Pairwise NA**
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes
    - w/ attributes
  - Optimal transport-based
    - w/o attributes

- **Collective NA**
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes

- **Higher-Order NA**
  - Consistency-based
    - Single-level
    - Multilevel
  - w/o attributes
  - w/ attributes

- **Related Tasks**
  - Entity alignment
    - Non-GNN based
    - GNN-based
  - Cross-layer inference
  - Cross-network transformation
Collective Network Alignment

• Goal: to find alignment across multiple networks

• Possible solution
  • Find pairwise alignment
  • Then combine
  • Transitivity constraint may be violated

• Problem setting:
  • Given: more than two networks $\mathcal{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_m\}$
  • Find: alignment across $\mathcal{G}_i, \mathcal{G}_j$ ($i, j = 1, \cdots, m$) jointly
Multiple Anonymized Social Networks Alignment

• Goal: to find anchor links/alignment across multiple networks without attributes
• Key challenge: how to preserve transitivity property
UMA – Unsupervised Pairwise Alignment

• Key idea: to minimize the alignment inconsistency
  • I.e., the number of non-shared edges between those mapped from $G^{(i)}$ and those in $G^{(j)}$

• Mathematical formulation

$$\bar{T}^{(i,j)} = \arg \min_{T^{(i,j)}} \left\| (T^{(i,j)})^\top S^{(i)} T^{(i,j)} - S(j) \right\|_F^2$$

s.t.  
$$T^{(i,j)} \in \{0, 1\} |U^{(i)}| \times |U^{(j)}|,$$

$$T^{(i,j)} 1_{|U^{(j)}|} \preceq 1_{|U^{(i)}|},$$

$$T^{(i,j)}^\top 1_{|U^{(i)}|} \preceq 1_{|U^{(j)}|}.$$  

one-to-one mapping constraints

• $S^{(i)}, S^{(j)}$: adjacency matrices of networks $G^{(i)}$ and $G^{(j)}$
• $T^{(i,j)}$: alignment matrix from $G^{(i)}$ to $G^{(j)}$

UMA – Transitivity Penalties

• Measure the number of inconsistent edges between the mapped from $\mathcal{G}^{(i)} \rightarrow \mathcal{G}^{(j)} \rightarrow \mathcal{G}^{(k)}$ and $\mathcal{G}^{(i)} \rightarrow \mathcal{G}^{(k)}$

• Mathematical formulation

$$ C(\{\mathcal{G}^{(i)}, \mathcal{G}^{(j)}, \mathcal{G}^{(k)}\}) = \left\| (T^{(j,k)})^\top (T^{(i,j)})^\top S^{(i)} T^{(i,j)} T^{(j,k)} - (T^{(i,k)})^\top S^{(i)} T^{(i,k)} \right\|_F^2 $$

• Extension to $n$ ($n \geq 3$) networks

$$ C(\{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(n)}\}) = \sum_{\forall \{\mathcal{G}^{(i)}, \mathcal{G}^{(j)}, \mathcal{G}^{(k)}\} \subseteq \{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(n)}\}} C(\{\mathcal{G}^{(i)}, \mathcal{G}^{(j)}, \mathcal{G}^{(k)}\}) $$

**UMA – Optimization Problem**

- **Objective:** to minimize the alignment inconsistency and transitivity penalties simultaneously

- **Mathematical formulation**

\[
\begin{align*}
\mathbf{T}^{(i,j)}, \mathbf{T}^{(j,k)}, \mathbf{T}^{(k,i)} \\
= \arg \min_{\mathbf{T}^{(i,j)}, \mathbf{T}^{(j,k)}, \mathbf{T}^{(k,i)}} & \| (\mathbf{T}^{(i,j)})^\top \mathbf{S}^{(i)} \mathbf{T}^{(i,j)} - \mathbf{S}^{(j)} \|_F^2 \\
+ & \| (\mathbf{T}^{(j,k)})^\top \mathbf{S}^{(j)} \mathbf{T}^{(j,k)} - \mathbf{S}^{(k)} \|_F^2 + \| (\mathbf{T}^{(k,i)})^\top \mathbf{S}^{(k)} \mathbf{T}^{(k,i)} - \mathbf{S}^{(i)} \|_F^2 \\
+ & \alpha \| (\mathbf{T}^{(j,k)})^\top (\mathbf{T}^{(i,j)})^\top \mathbf{S}^{(i)} \mathbf{T}^{(i,j)} - \mathbf{T}^{(k,i)} \mathbf{S}^{(i)} (\mathbf{T}^{(k,i)})^\top \|_F^2
\end{align*}
\]

\[
\text{s.t.} \quad \mathbf{T}^{(i,j)} \in \{0, 1\}^{\mathcal{U}^{(i)} \times \mathcal{U}^{(j)}}, \quad \mathbf{T}^{(j,k)} \in \{0, 1\}^{\mathcal{U}^{(j)} \times \mathcal{U}^{(k)}} \\
\mathbf{T}^{(k,i)} \in \{0, 1\}^{\mathcal{U}^{(k)} \times \mathcal{U}^{(i)}}
\]

- Alignment inconsistency
- Transitivity penalties
- One-to-one constraints
- Relaxations
- Linear constraint + L1 norm

---

Uma – Transitive Network Matching

• Goal: to solve for binary variable $x_{l,m}^{(i,j)}$ indicating whether node $u_l$ in $G^{(i)}$ is aligned with node $u_m$ in $G^{(j)}$

• Optimization problem
  • Select high scores in alignment
  • One-to-one constraint
  • Transitivity constraint

\[
\max_{x^{(i,j)}, x^{(j,k)}, x^{(k,i)}} \sum_{l,m} x_{l,m}^{(i,j)} T^{(i,j)}(l,m) + \sum_{l,m} x_{l,m}^{(j,k)} T^{(j,k)}(l,m),
\]

s.t.
\[
\sum_{u_l^{(j)}} x_{l,m}^{(i,j)} \leq 1, \sum_{u_l^{(i)}} x_{l,o}^{(i,k)} \leq 1, \forall u_l^{(i)} \in U^{(i)},
\]
\[
\sum_{u_l^{(j)}} x_{m,l}^{(j,k)} \leq 1, \sum_{u_l^{(k)}} x_{m,o}^{(j,k)} \leq 1, \forall u_m^{(j)} \in U^{(j)},
\]
\[
\sum_{u_l^{(i)}} x_{o,l}^{(k,i)} \leq 1, \sum_{u_l^{(k)}} x_{o,m}^{(k,i)} \leq 1, \forall u_o^{(k)} \in U^{(k)},
\]
\[
x_{l,m}^{(i,j)} + x_{m,o}^{(j,k)} + x_{o,l}^{(k,i)} \neq 2, \forall l \in \{1, 2, \ldots, |U^{(i)}|\}, m \in \{1, 2, \ldots, |U^{(j)}|\}, o \in \{1, 2, \ldots, |U^{(k)}|\},
\]
\[
x_{l,m}^{(i,j)} \in \{0, 1\}, \forall u_l^{(i)} \in U^{(i)}, u_m^{(j)} \in U^{(j)}.
\]
\[
x_{m,o}^{(j,k)} \in \{0, 1\}, \forall u_m^{(j)} \in U^{(j)}, u_o^{(k)} \in U^{(k)}.
\]
\[
x_{o,l}^{(k,i)} \in \{0, 1\}, \forall u_o^{(k)} \in U^{(k)}, u_l^{(i)} \in U^{(i)}.
\]

**UMA – Experimental Results**

- **Dataset:** Stack Overflow, Super User and Programmers
- **Alignment performance**

---

**References:**

COSNET: Connecting Social Networks with Local and Global Consistency

• Intuitions: binary classification over node pairs
  • Instances: node pairs $X = \{x_i\}$
  • Labels: $Y = \{y_i\}$, $y_i = 1$ if $x_i$ refers to same node, otherwise 0

• Factors considered:
  • Node feature consistency (e.g., user profiles)
  • Structural consistency
  • Global consistency (i.e., transitivity constraints)

COSNET – Node Feature Consistency

• Intuition: to encode the feature similarity for $x_i$

• Formulation:

$$E_l(Y, X) = \sum_i w_l^T g_l(x_i, y_i)$$

• $g_l(x_i, y_i)$ is a vector-valued feature function
  • Encodes the user profile similarity for node pair $x_i$
• $w_l$ is the model parameter

COSNET – Structural Consistency

• Intuition:
  • If two nodes are aligned, their neighbors are likely to be aligned

• Matching graph $MG = (V_{MG}, E_{MG})$
  • Same as Kronecker product graph

• Pairwise formulation:

$$E_e(Y, X) = \sum_{\langle x_i, x_j \rangle \in E_{MG}} w_e^T f_e(y_i, y_j)$$

$$f_e(y_i, y_j) = \begin{cases} 
(1, 0, 0)^T & \text{if } y_i = y_j = 0 \\
(0, 1, 0)^T & \text{if } y_i + y_j = 1 \\
(0, 0, 1)^T & \text{if } y_i = y_j = 1
\end{cases}$$

**Definition 2 (Global Inconsistency).** Given a set of social networks $\mathbf{G}$, a set of user pairs $X$ and the corresponding labels $Y$, if there exists a sequence of user pairs $(x_{i_1}, x_{i_2}, \ldots, x_{i_n})$, such that

$$\forall i = i_1, i_2, \ldots, i_n, y_i = 1$$

and

$$\forall k = 1, 2, \ldots, n - 1, V^2_{i_k} = V^1_{i_{k+1}}$$

and

For the pair $\langle V^2_{i_n}, V^1_1 \rangle$, the corresponding label $y_j = 0$

then we say that the assigned labels $Y$ causes global inconsistency given $\mathbf{G}$ and $X$. 

COSNET – Global Consistency

• Triadic closure in the matching graph

• Formulation:

\[ E_t(Y, X) = \sum_{c \in T_{MG}} w_t^T f_t(Y_c) \]

\[ f_t(y_i, y_j) = \begin{cases} 
(1, 0, 0, 0)^\top & \text{if } |Y_c| = 0 \\
(0, 1, 0, 0)^\top & \text{if } |Y_c| = 1 \\
(0, 0, 1, 0)^\top & \text{if } |Y_c| = 2 \\
(0, 0, 0, 1)^\top & \text{if } |Y_c| = 3 
\end{cases} \]
COSNET – Model Learning

• Objective function:

\[ E(Y, X) = \sum_{x_i \in V_{MG}} w_l^T g_l(x_i, y_i) + \sum_{\langle x_i, x_j \rangle \in E_{MG}} w_e^T f_e(y_i, y_j) + \sum_{c \in T_{MG}} w_c^T f_c(Y_c) \]

• Define distance of two matching configurations \( Y \) and \( Y' \)

\[ \Delta(Y, Y') = \sum_{x_i \in V_{MG}} \delta_l(y_i, y_i') + \sum_{c \in T_{MG}} \delta_c(Y_c, Y_c') + \sum_{\langle x_i, x_j \rangle \in E_{MG}} \delta_e(\langle y_i, y_j \rangle, \langle y'_i, y'_j \rangle) \]

Hamming distance
COSNET – Model Learning

• By max-margin theory:

\[
\min_W \frac{1}{2} ||W||^2 + \mu \xi \\
\text{s.t.} \quad E(\hat{Y}, X; W) \leq E(Y, X; W) - \Delta(Y, \hat{Y}) + \xi
\]

• \(\hat{Y}, Y\): input labeled configuration and learned configuration
• \(W = (w_l, w_e, w_t)\): model parameters
• \(\xi\): slack variable to handle non-separable data
• \(\mu\): trade-off between the maximum margin & error penalty
• Constraint: distance between the energy of \(\hat{Y}, Y\) at least \(\Delta(Y, \hat{Y})\)

COSNET – Model Learning

• The original problem is intractable.
• Use Lagrangian relaxation for dual decomposition

\[
\min_{W, \lambda} \frac{1}{2} \|W\|^2 + \mu (E(\hat{Y}, X; W) - \max_{\lambda} L(Y, X, \lambda; W))
\]

\[
\text{s.t. } \sum_{y_i \in Y_i} \lambda^f_i = 0, \ \forall f \in \mathcal{F}
\]

• \(f \in \mathcal{F}\): factor functions
• \(\lambda\): Lagrange multipliers

• Convex and non-differentiable
• Solution: projected sub-gradient method

COSNET – Public Dataset

• Data statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Network</th>
<th>#Users</th>
<th>#Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNS</td>
<td>Twitter</td>
<td>40,171,624</td>
<td>1,468,365,182</td>
</tr>
<tr>
<td></td>
<td>LiveJournal</td>
<td>3,017,286</td>
<td>87,037,567</td>
</tr>
<tr>
<td></td>
<td>Flickr</td>
<td>215,495</td>
<td>9,114,557</td>
</tr>
<tr>
<td></td>
<td>Last.fm</td>
<td>136,420</td>
<td>1,685,524</td>
</tr>
<tr>
<td></td>
<td>MySpace</td>
<td>854,498</td>
<td>6,489,736</td>
</tr>
<tr>
<td>Academia</td>
<td>LinkedIn</td>
<td>2,985,414</td>
<td>25,965,384</td>
</tr>
<tr>
<td></td>
<td>ArnetMiner</td>
<td>1,053,188</td>
<td>3,916,907</td>
</tr>
<tr>
<td></td>
<td>VideoLectures</td>
<td>11,178</td>
<td>786,353</td>
</tr>
</tbody>
</table>

• Link: https://www.aminer.cn/cosnet

COSNET – Experimental Results

• Connecting social media sites
  • Twitter, LiveJournal, Last.fm, Flickr, MySpace

COSNET – Experimental Results

- Connecting Aminer with LinkedIn and VideoLectures

Overview of Part I

Part I: Recent Network Alignment (NA) Algorithms

**Pairwise NA**
- Consistency-based
  - w/o attributes
  - w/ attributes
- Embedding-based
  - w/o attributes
  - w/ attributes
- Optimal transport-based
  - w/o attributes

**Collective NA**
- Consistency-based
  - w/o attributes
  - w/ attributes
- **Embedding-based**
  - w/o attributes

**Higher-Order NA**
- Consistency-based
  - Single-level
  - Multilevel

**Related Tasks**
- Entity alignment
  - Non-GNN based
  - GNN-based
- Cross-layer inference
- Cross-network transformation
Embedding-Based Collective Network Alignment

- Goal: to learn node embeddings that can infer alignment in the embedding space
Cross-Network Embedding for Multi-Network Alignment

• Motivations: networks heterogeneity
  • Different networks may own different semantic meanings;
  • Same node may have distinct embeddings in different networks

• Goal: to learn node embeddings for multiple network alignment

• Key question: how to capture the commonness among anchor node counterparts and specific semantics in different networks?

CrossMNA – Cross Network Embedding

• Key idea: split node embedding into two components

\[ \mathbf{v}_i^k = \mathbf{Wu}_i + \mathbf{r}_k \]

• Intra-vector \( \mathbf{v}_i^k \): captures structural information in a network
• Inter-vector \( \mathbf{u}_i \): captures the commonness of anchor node
• Network vector \( \mathbf{r}_k \): captures network-specific semantics

CrossMNA – Experimental Results

• Multiple network alignment

Twitter dataset

Arxiv dataset

Precision@\(\alpha\) vs. \(\alpha\)  Precision@30 vs. training ratio

CrossMNA – Experimental Results

• Multiple network link prediction

<table>
<thead>
<tr>
<th>Dataset</th>
<th>arXiv</th>
<th>SacchCere</th>
<th>Twitter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30% 50% 80%</td>
<td>30% 50% 80%</td>
<td>30% 50% 80%</td>
</tr>
<tr>
<td>DeepWalk</td>
<td>87.86 94.41 98.12</td>
<td>69.20 73.96 78.29</td>
<td>57.09 59.96 63.74</td>
</tr>
<tr>
<td>LINE</td>
<td>75.05 85.73 94.75</td>
<td>60.54 65.05 68.87</td>
<td>53.12 52.97 53.22</td>
</tr>
<tr>
<td>node2vec</td>
<td>88.06 94.57 97.11</td>
<td>71.15 76.33 80.16</td>
<td>56.84 61.22 65.93</td>
</tr>
<tr>
<td>PMNE</td>
<td>90.12 94.47 95.24</td>
<td>77.61 79.85 81.35</td>
<td>61.12 70.72 75.91</td>
</tr>
<tr>
<td>MELL</td>
<td>93.51 96.30 98.84</td>
<td>76.18 79.92 81.21</td>
<td>70.64 75.89 79.84</td>
</tr>
<tr>
<td>CrossMNA</td>
<td>96.46 97.53 99.19</td>
<td>76.88 81.12 82.59</td>
<td>75.85 80.48 85.29</td>
</tr>
</tbody>
</table>

**Observation:** CrossMNA performs better due to transmitting complementary information across networks.

CrossMNA – Experimental Results

• Scalability: memory usage

Observation: CrossMNA has less memory usage than other baseline methods.
Overview of Part I

Part I: Recent Network Alignment (NA) Algorithms

Pairwise NA
- Consistency-based
  - w/o attributes
  - w/ attributes
- Embedding-based
  - w/o attributes
  - w/ attributes
- Optimal transport-based
  - w/o attributes

Collective NA
- Consistency-based
  - w/o attributes
  - w/ attributes
- Embedding-based
  - w/o attributes

Higher-Order NA
- Consistency-based
  - Single-level
  - Multilevel
- Optimal transport-based
  - w/o attributes

Related Tasks
- Entity alignment
  - Non-GNN based
  - GNN-based
- Cross-layer inference
- Cross-network transformation
Higher-order Network Alignment

• Higher-order network mining:
  • Involves higher-order structures, instead of edges

• Motivations:
  • Traditional approaches (e.g., NetAlign) aim to maximize # of conserved edges (overlaps/squares).
  • Leverage higher-order structures exist in networks (e.g., motifs, clusters, etc.).

• Single-level: use higher-order structures to align nodes
• Multilevel: to align both nodes and clusters at multi-level
Triangular Alignment (TAME)

- Network motifs: connected subgraphs that occurs with significantly higher frequency
  - 3rd-order: 3-node line, triangle
  - kth-order: k-node star, etc.

- Objective: to maximize # of aligned substructures

**TAME – Formulation #1**

- **Binary quadratic program in NetAlign**

\[
\begin{align*}
\text{maximize} & \quad (1 - \alpha)w^T x + \frac{\alpha}{2} x^T S x \\
\text{subject to} & \quad Cx \leq 1_{|V_G|+|V_H|} \\
& \quad x(i'i') \in \{0, 1\}.
\end{align*}
\]

- **Higher-order extension**

\[
\begin{align*}
\text{maximize} & \quad (1 - \alpha)w^T x + \frac{\alpha}{m!} (\mathcal{T}_H \otimes \mathcal{T}_G)x^m \\
\text{subject to} & \quad Cx \leq 1_{|V_G|+|V_H|} \\
& \quad x(i'i') \in \{0, 1\}.
\end{align*}
\]

\[
x^T (\mathcal{T}_H \otimes \mathcal{T}_G)x^{m-1} = (\mathcal{T}_H \otimes \mathcal{T}_G)x^m
\]

- \(\mathcal{T}_H\) and \(\mathcal{T}_G\): the motif-tensors associated with a \(m\)-node motif in both graphs \(G\) and \(H\)

- \(\Delta_{H \times G} = \Delta_H \otimes \Delta_G\): Kronecker product of triangle tensors

- Counts # of conserved triangles

TAME – Formulation #2

• Relaxed formulation
  • Remove one-to-one constraint and relax $x$ to be any reals
  • Add a 2-norm constraint on $x$ to make it bounded

$$\max_x \quad (\Delta_{H \times G} x)^3$$
subject to \quad $\|x\| = 1.$

$\rightarrow$ Tensor eigenvector problem

• The classic SS-HOPM is costly to solve it.

• Implicit kernel for computing tensor-vector products

$$(\Delta_{H \times G} x^2)_{ii'}$$
$$= \sum_{jj',kk'} \Delta_{H \times G}(ii',jj',kk')x(jj')x(kk')$$
$$= \sum_{jj',kk'} \Delta_G(i,j,k) \Delta_H(ii',jj',kk')X(jj')X(kk')$$
$$= \sum_{jj',kk'} \Delta_G(i,j,k) \sum_{jj'} X(jj') \sum_{kk'} \Delta_H(ii',jj',kk')X(kk')$$

$\Delta_{H \times G} x^2(ii') = 2 \sum_{(j,k) \in N_{\Delta_G}(i)} \sum_{(j',k')} X(jj')X(kk') + X(jj')X(kk').$$

TAME – Algorithm

• Key ideas:
  • To use implicit tensor-kernel product \( \tilde{x} = \Delta_{H\times G} x^2 \) for \( \Delta_{H\times G} x^3 = x^T \tilde{x} \)
  • SS-HOPM main loop computes topological similarity matrices
  • A score function to solve a bipartite max-weight matching

To encode integer constraint of \( X \) and one-to-one mapping constraint

TAME – Experimental Results

• Alignment quality on yeast vs. human dataset

Observation: TAME performs closely to the best method in preserving the # of conserved edges

TAME – Experimental Results

• Metric: # of conserved triangles

Observation: TAME ranks the highest in terms of the number of conserved triangles

Multilevel Network Alignment

- **Goals:** to find node correspondence as well as the correspondence among clusters at different levels
- **Motivation:**
  - Networks exhibit hierarchical cluster-within-clusters structure

Moana – Challenges

• C1: Alignment accuracy

• Errors propagate through levels

• C2: Scalability
  Better than quadratic?

Moana – Problem Definition

• Given:
  • (1) adjacency matrices $\bar{A}_1, \bar{B}_1$ of two undirected networks;
  • (2) a sparse prior alignment preference $H_1$;
  • (3) the number of levels $L \geq 2$ of interests.

• Find: a set of alignment matrices $S_l$ at level-$l$, $l = 1, \ldots, L$
  • where $S_1$ indicates the alignment at the node level

Moana Formulation: Multilevel Optimization

• Generic strategy
  • coarsening $\rightarrow$ alignment $\rightarrow$ interpolation

• Alignment interpolations
  • Bilinear interpolations by $P_l \in \mathbb{R}^{p_l \times n_1}$, $Q_l \in \mathbb{R}^{q_l \times n_2}$ ($p_l \leq n_1, q_l \leq n_2$)
  • w.l.o.g., $S_1 = Q_1^T S_2 P_1$ between level-1 & level-2

Moana Formulation: Multilevel Optimization

- Multilevel alignment formulation
  
  **Level-1:** \[
  \min_{\mathbf{s}_1} \alpha \mathbf{s}_1^T (\mathbf{I} - \mathbf{A}_1 \otimes \mathbf{B}_1) \mathbf{s}_1 + (1 - \alpha) \| \mathbf{s}_1 - \mathbf{h}_1 \|_2^2
  \]
  
  **Level-2:** \[
  \min_{\mathbf{s}_2} \alpha \mathbf{s}_2^T (\mathbf{I} - \mathbf{A}_2 \otimes \mathbf{B}_2) \mathbf{s}_2 + (1 - \alpha) \| \mathbf{s}_2 - \mathbf{h}_2 \|_2^2
  \]

  If \( \mathbf{P}_1 \mathbf{P}_1^T = \mathbf{I} \) and \( \mathbf{Q}_1 \mathbf{Q}_1^T = \mathbf{I} \)

- \( \mathbf{A}_2 = \mathbf{P}_1 \mathbf{A}_1 \mathbf{P}_1^T, \mathbf{B}_2 = \mathbf{Q}_1 \mathbf{B}_1 \mathbf{Q}_1^T \) and \( \mathbf{H}_2 = \mathbf{Q}_1 \mathbf{H}_1 \mathbf{P}_1^T \)
- same properties (e.g., convexity) and algorithm as FINAL-P
- ‘good’ (semi-) orthogonal \( \mathbf{P}_1, \mathbf{Q}_1 \) make \( \mathbf{A}_2, \mathbf{B}_2 \) well-represented

Moana Formulation: Perfect Interpolation

• Denote $S_l^*, S_{l+1}^*$ are optimal solutions at level-$l$ and level-$(l + 1)$

• Perfect interpolation (to address error propagation):

  Interpolation from the optimal alignment matrix at level-$(l + 1)$ is equal to that at level-$l$

• If $P_l, Q_l$ ($l = 1, \cdots, L - 1$) are orthogonal
• Then $S_l^* = Q_l^T S_{l+1}^* P_l$

Moana – Coarsening Algorithm

• Generic strategy
  • Coarsening $\rightarrow$ alignment $\rightarrow$ interpolation

• Network coarsening by $P_l, Q_l$
  • $A_{l+1} = P_l A_l P_l^T, B_{l+1} = Q_l B_l Q_l^T$

• Requirements on $P_l, Q_l$
  • Perfect interpolation: they are orthogonal matrix
  • Efficient computation: they are sparse matrix
  • Informative coarsening: they can uncover hierarchical cluster-within-clusters structures

Moana – Coarsening Algorithm

• Multiresolution matrix factorization

\[
\begin{align*}
S_{L-1} & \rightarrow \cdots \rightarrow S_2 & \rightarrow S_1 & \rightarrow A_1 & \rightarrow P_1^T & \rightarrow P_2 & \rightarrow \cdots & \rightarrow P_{L-1}^T & \rightarrow P_L \rightarrow \tilde{A}_L \\
S_{L-1} & \rightarrow \cdots \rightarrow S_2 & \rightarrow S_1 & \rightarrow B_1 & \rightarrow Q_1^T & \rightarrow Q_2 & \rightarrow \cdots & \rightarrow Q_{L-1}^T & \rightarrow Q_L \rightarrow \tilde{B}_L
\end{align*}
\]

• Coarsening procedure

\[
\begin{align*}
P_{L-1} & \cdots & P_2 P_1 A_1 P_1^T P_2^T & \cdots & P_{L-1}^T & = A_L & \rightarrow \tilde{A}_L \\
Q_{L-1} & \cdots & Q_2 Q_1 B_1 Q_1^T Q_2^T & \cdots & Q_{L-1}^T & = B_L & \rightarrow \tilde{B}_L
\end{align*}
\]

• \( \mathcal{S}(\mathcal{S}_{B_l}, \mathcal{S}_{A_l}) \) indicates the alignment among clusters at the \( l \)-th granularity

Moana – Alignment Algorithm

• Generic strategy
  • coarsening → alignment → interpolation

• Alignment across the coarsest networks

\[
\tilde{S}_L = \alpha \begin{bmatrix} \tilde{B}_{L_1} & 0 \\ 0 & \tilde{B}_{L_2} \end{bmatrix} \begin{bmatrix} \tilde{S}_{L_1} \\ \tilde{S}_{L_2} \end{bmatrix} \begin{bmatrix} \tilde{A}_{L_1} & 0 \\ 0 & \tilde{A}_{L_2} \end{bmatrix} + (1 - \alpha) \begin{bmatrix} \tilde{H}_{L_1} & \tilde{H}_{L_2} \\ \tilde{H}_{L_3} & \tilde{H}_{L_4} \end{bmatrix}
\]

\[
\begin{align*}
\tilde{S}_{L_1} &= \alpha \tilde{B}_{L_1} \tilde{S}_{L_1} \tilde{A}_{L_1} + (1 - \alpha) \tilde{H}_{L_1} \\
\tilde{S}_{L_2} &= \alpha \tilde{B}_{L_1} \tilde{S}_{L_2} \tilde{A}_{L_2} + (1 - \alpha) \tilde{H}_{L_2} \\
\end{align*}
\]

block-wise computation

\[
\begin{align*}
\tilde{S}_{L_3} &= \alpha \tilde{B}_{L_2} \tilde{S}_{L_3} \tilde{A}_{L_1} + (1 - \alpha) \tilde{H}_{L_3} \\
\tilde{S}_{L_4} &= (1 - \alpha) \left( I - \alpha \tilde{A}_{L_2} \otimes \tilde{B}_{L_3} \right)^{-1} \tilde{H}_{L_4}
\end{align*}
\]

• Alignment at finer levels
  • perfect interpolations: \( S_l = Q_l^T S_{l+1} P_l \)

Moana – Experimental Setups

• Datasets
  • Gr-Qc network vs. its permutation (nodes: 5,241 vs. 5,241)
  • Google+ vs. its permutation (nodes: 23,628 vs. 23,628)
  • Amazon co-purchasing networks (nodes: 74,596 vs. 66,951)
  • ACM vs DBLP coauthor networks (nodes: 9,872 vs. 9,916)

Moana – Experimental Results

Observations: (1) the performance of Moana is close to FINAL-P; (2) Moana outperforms all other methods.

Moana – Experimental Results

**Observation:** Moana achieves a good performance in cluster alignment at different levels.

Moana – Experimental Results

**Observation:** Moana can unveil meaningful alignment of clusters at different granularities.

Moana – Experimental Results

Observation: (1) Moana scales linearly w.r.t. the number of edges; (2) Moana scales linearly w.r.t. the number of nonzero elements in $H_1$.

Overview of Part I

Part I: Recent Network Alignment (NA) Algorithms

- Pairwise NA
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes
    - w/ attributes
  - Optimal transport-based
    - w/o attributes

- Collective NA
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes

- Higher-Order NA
  - Consistency-based
    - Single-level
    - Multilevel

- Related Tasks
  - Entity alignment
    - Non-GNN based
    - GNN-based
  - Cross-layer inference
  - Cross-network transformation
Entity Alignment

• Goal: to link entities among multiple knowledge graphs
• Problem Definition:
  • Given KGs \( \{ KG_i \mid KG_i = (E_i, R_i, T_i) \} \) and seed alignment \( \mathcal{L} \);
  • Find all the aligned entities
Iterative Entity Alignment via Joint Knowledge Embeddings

• Key components:
  • Knowledge embedding: TransE, PTransE
  • Joint embedding: translation-based, linear transformation
  • Iterative alignment: adding newly aligned entities

IITransE – Knowledge Embeddings

• TransE: relations as translating vectors

\[ E(h, r, t) = \|h + r - t\| \]

• Loss function:

\[ L(h, r, t) = \sum_{(h', r', t') \in T^-} [\gamma + E(h, r, t) - E(h', r', t')]+ \]

• Negative samples:

\[ T^- = \{(h', r, t)|h' \in E\} \cup \{(h, r, t')|t' \in E\} \cup \{(h, r', t)|r' \in R\}, \quad (h, r, t) \in T. \]

• PTransE: to encode multi-step relation path

\[ E(p, r) = \|p - r\| = \|p - (t - h)\| = E(h, p, t) \]

ITransE – Joint Embeddings

• Key idea: to join embeddings in a unified space

• Translation-based model:
  • Key idea: view alignment as a special relation
  • Formulation: given $e_1 \in E_1, e_2 \in E_2 \rightarrow e_1 + r^{(E_1 \rightarrow E_2)} \approx e_2$

$$E(e_1, e_2) = ||e_1 + r^{(E_1 \rightarrow E_2)} - e_2||.$$ 

• Linear transformation model:
  • Key idea: embedding space can be transformed linearly
  • Formulation: transformation matrix $M^{(E_1 \rightarrow E_2)}$

$$E(e_1, e_2) = ||M^{(E_1 \rightarrow E_2)}e_1 - e_2||.$$ 

ITransE – Iterative Alignment

• Key idea: iteratively adding newly aligned entities

• Soft alignment:
  • Reliability scores of newly aligned entities
    \[ R(e_1, e_2) = \sigma \left( k(\theta - E(e_1, e_2)) \right) \]

  • Score function for soft alignment
    \[ I_S = \sum_{(e_1, e_2) \in M} R(e_1, e_2)(\mathcal{H}(e_1, e_2) + \mathcal{H}(e_2, e_1)), \]
    \[ \mathcal{H}(e_1, e_2) = \sum_{(e_1, r, t)} U(e_2, r, t) + \sum_{(h, r, e_1)} U(h, r, e_2), \]

  • Limit # of newly aligned entities to a threshold in each alignment procedure

ITransE – Experimental Results

• Dataset: DFB1, DFB2, DFB3 from FB15K
• Entity alignment performance
  • ITransE: iterative alignment w/ TransE
  • IPTransE: iterative alignment w/ PTransE

<table>
<thead>
<tr>
<th>Metric</th>
<th>DFB-1 Hits@1</th>
<th>Hits@10</th>
<th>Mean Rank</th>
<th>DFB-2 Hits@1</th>
<th>Hits@10</th>
<th>Mean Rank</th>
<th>DFB-3 Hits@1</th>
<th>Hits@10</th>
<th>Mean Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTransE (LT)</td>
<td>38.9</td>
<td>61.0</td>
<td>237.7</td>
<td>12.3</td>
<td>33.8</td>
<td>419.2</td>
<td>6.5</td>
<td>22.0</td>
<td>699.8</td>
</tr>
<tr>
<td>MTransE (TB)</td>
<td>13.6</td>
<td>35.1</td>
<td>547.7</td>
<td>13.9</td>
<td>35.4</td>
<td>675.7</td>
<td>4.5</td>
<td>16.1</td>
<td>1255.5</td>
</tr>
<tr>
<td>TransE + PS</td>
<td>61.9</td>
<td>79.2</td>
<td>105.2</td>
<td>41.1</td>
<td>67.0</td>
<td>154.9</td>
<td>12.2</td>
<td>34.6</td>
<td>431.9</td>
</tr>
<tr>
<td>ITransE (HA)</td>
<td>62.6</td>
<td>78.9</td>
<td>100.0</td>
<td>41.2</td>
<td>66.9</td>
<td>151.9</td>
<td>12.3</td>
<td>33.7</td>
<td>432.3</td>
</tr>
<tr>
<td>ITransE (SA)</td>
<td>67.1</td>
<td>83.1</td>
<td>80.1</td>
<td>57.7</td>
<td>77.7</td>
<td>109.3</td>
<td>16.2</td>
<td>40.9</td>
<td>367.2</td>
</tr>
<tr>
<td>PTransE + PS</td>
<td>65.8</td>
<td>83.4</td>
<td>62.9</td>
<td>46.3</td>
<td>72.1</td>
<td>96.8</td>
<td>15.8</td>
<td>40.2</td>
<td>346.9</td>
</tr>
<tr>
<td>IPTransE (HA)</td>
<td>66.1</td>
<td>83.3</td>
<td>59.1</td>
<td>46.2</td>
<td>72.6</td>
<td>94.2</td>
<td>15.1</td>
<td>39.7</td>
<td>337.6</td>
</tr>
<tr>
<td>IPTransE (SA)</td>
<td>71.7</td>
<td>86.5</td>
<td>49.0</td>
<td>63.5</td>
<td>82.2</td>
<td>67.5</td>
<td>20.4</td>
<td>47.4</td>
<td>281.0</td>
</tr>
</tbody>
</table>

Observations:
• IPTransE performs better than ITransE
• Soft alignment performs better than hard alignment

ITransE – Experimental Results

• Effectiveness of soft alignment strategy

Observation: the performance of all methods increase with iterations.

Knowledge Graph Alignment via Graph Convolutional Networks

• Key idea: use GCNs to embed entities where aligned entities are expected to be as close as possible.

• Assumptions:
  • Equivalent entities tend to have similar attributes
  • Equivalent entities are neighbored by other equivalent entities

• Embedding framework:

\[
\begin{align*}
[H_s^{(l+1)}; H_a^{(l+1)}] \\
= \sigma \left( \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} [H_s^{(l)} W_s^{(l)}; H_a^{(l)} W_a^{(l)}] \right)
\end{align*}
\]

GCN-Align – Construct Adjacency Matrix

• KGs are relational multi-graphs (i.e., typed relations)
• Key idea: two measures on relations

Relation functionality: \( \text{fun}(r) = \frac{\#\text{Head}\_\text{Entities}\_\text{of}\_r}{\#\text{Triples}\_\text{of}\_r} \)

Inverse functionality: \( \text{ifun}(r) = \frac{\#\text{Tail}\_\text{Entities}\_\text{of}\_r}{\#\text{Triples}\_\text{of}\_r} \)

• Edge weight: influence of \( i \)-th entity over \( j \)-th entity

\[
a_{ij} = \sum_{\langle e_i, r, e_j \rangle \in G} \text{ifun}(r) + \sum_{\langle e_j, r, e_i \rangle \in G} \text{fun}(r)
\]

GCN-Align – Alignment Prediction

- Model training:
  - Margin-based rank loss for both $h_s$ and $h_a$
  - $h_s$: structure embedding
  - $h_a$: attribute embedding
- Small distance for aligned entities for prediction

$$D(e_i, v_j) = \beta \frac{\|h_s(e_i) - h_s(v_j)\|_1}{d_s} + (1 - \beta) \frac{\|h_a(e_i) - h_a(v_j)\|_1}{d_a}$$

- $d_s, d_a$: dimensions of structure and attribute embedding
- $\beta$: hyperparameter balancing importance of two embeddings
- For each entity $e_i$, return a list of candidate entities in $\text{KG}_2$

GCN-Align – Experimental Results

• Datasets: DBP15K from DBpedia with different languages

| DBP15K_{JA-EN} |  |  |  |
|----------------|------------------|------------------|
|               | JA → EN          | EN → JA          |
|               | Hits@1 | Hits@10 | Hits@50 | Hits@1 | Hits@10 | Hits@50 |
| *JE           | 18.92   | 39.97   | 54.24   | 17.80   | 38.44   | 52.48   |
| *MTransE      | 27.86   | 57.45   | 75.94   | 23.72   | 49.92   | 67.93   |
| *JAPE         | 33.10   | 63.90   | 80.80   | 29.71   | 56.28   | 73.84   |
|               | 34.27   | 66.39   | 83.61   | 31.40   | 60.80   | 78.51   |
|               | 36.25   | 68.50   | 85.35   | 38.37   | 67.27   | 82.65   |
| JAPE’         | 29.80   | 60.61   | 80.03   | 25.34   | 53.36   | 71.94   |
|               | 29.35   | 63.31   | 82.76   | 26.37   | 57.35   | 76.87   |
|               | 31.06   | 64.11   | 81.57   | 32.45   | 62.21   | 79.08   |
| GCN           | 38.21   | 72.49   | 82.69   | 36.90   | 68.50   | 79.51   |
|               | 39.91   | 74.46   | 86.10   | 38.42   | 71.81   | 83.72   |

| DBP15K_{FR-EN} |  |  |  |
|----------------|------------------|------------------|
|               | FR → EN          | EN → FR          |
|               | Hits@1 | Hits@10 | Hits@50 | Hits@1 | Hits@10 | Hits@50 |
| *JE           | 15.38   | 38.84   | 56.50   | 14.61   | 37.25   | 54.01   |
| *MTransE      | 24.41   | 55.55   | 74.41   | 21.26   | 50.60   | 69.93   |
| *JAPE         | 29.55   | 62.18   | 79.36   | 25.40   | 56.55   | 74.96   |
|               | 29.63   | 64.55   | 81.90   | 26.55   | 60.30   | 78.71   |
|               | 32.39   | 66.68   | 83.19   | 32.97   | 65.91   | 82.38   |
| JAPE’         | 28.23   | 60.99   | 78.47   | 24.68   | 55.25   | 74.19   |
|               | 27.58   | 62.03   | 79.98   | 24.93   | 58.95   | 77.79   |
|               | 30.21   | 65.81   | 82.57   | 31.42   | 63.86   | 80.95   |
| GCN           | 36.51   | 73.42   | 85.93   | 36.08   | 72.37   | 85.44   |
|               | 37.29   | 74.49   | 86.73   | 36.77   | 73.06   | 86.39   |

Overview of Part I

Part I: Recent Network Alignment (NA) Algorithms

- Pairwise NA
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes
    - w/ attributes
  - Optimal transport-based
    - w/o attributes

- Collective NA
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes

- Higher-Order NA
  - Consistency-based
    - Single-level
    - Multilevel
  - Embedding-based
    - w/o attributes

- Related Tasks
  - Entity alignment
    - Non-GNN based
    - GNN-based
  - Cross-layer inference
  - Cross-network transformation
Multi-layered Networks

• An example of multi-layered networks

Cross-Layer Dependency Inference

• Given: a multi-layered network
  • Layer-layer dependency matrix $G$
  • Within-layer connectivity matrices $\mathcal{A} = \{A_1, \cdots, A_g\}$
  • Observed cross-layer dependency matrices $\mathcal{D} = \{D_{ij}\}$
• Find: true cross-layer dependency matrices $\{\tilde{D}_{ij}\}$
• To link different types of nodes (alignment links same)
  • $A_1$ for chemical network, etc.
  • $G(1,2) = 1$, $G(1,3) = 0$
  • $D_{12}$ are represented by solid arrows between $\mathcal{G}_1$ and $\mathcal{G}_2$

Fascinate – Formulation

• Key idea: as a collective collaborative filtering problem
  • Within-layer networks as user-user network, item-item similarity network, etc.
  • Cross-layer dependency as user-item ratings

• Optimization problem:

\[
J = \min_{F_i \geq 0(i=1,\ldots,g)} \sum_{i,j: \mathbf{G}(i,j)=1} \left\| \mathbf{W}_{i,j} \odot (\mathbf{D}_{i,j} - \mathbf{F}_i \mathbf{F}_j^t) \right\|_F^2
\]

\[+ \alpha \sum_{i=1}^{g} \text{tr}(\mathbf{F}_i^t(\mathbf{T}_i - \mathbf{A}_i)\mathbf{F}_i) + \beta \sum_{i=1}^{g} \| \mathbf{F}_i \|_F^2\]

C1: Matching Observed Cross-Layer Dependencies
C2: Node Homophily
C3: Regularization

Fascinate – Optimization Algorithm

• Block coordinate descent method
• For each $F_i$, use multiplicative update method

\[
\frac{\partial J_i}{\partial F_i} = 2 \sum_{j: G(i,j)=1} \left[-(W_{i,j} \odot W_{i,j} \odot D_{i,j})F_j + (W_{i,j} \odot W_{i,j} \odot (F_i F_j'))F_j \right] + \alpha T_i F_i - \alpha A_i F_i + \beta F_i
\]

\[
F_i(u,v) \leftarrow F_i(u,v) \sqrt{\frac{X(u,v)}{Y(u,v)}},\text{ where}
\]

\[
X = \sum_{j: G(i,j)=1} (W_{i,j} \odot W_{i,j} \odot D_{i,j})F_j + \alpha A_i F_i
\]

\[
Y = \sum_{j: G(i,j)=1} (W_{i,j} \odot W_{i,j} \odot (F_i F_j'))F_j + \alpha T_i F_i + \beta F_i
\]

Fascinate – Experimental Setups

• Datasets:

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of Layers</th>
<th># of Nodes</th>
<th># of Links</th>
<th># of CrossLinks</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOCIAL</td>
<td>3</td>
<td>125,344</td>
<td>214,181</td>
<td>188,844</td>
</tr>
<tr>
<td>BIO</td>
<td>3</td>
<td>35,631</td>
<td>253,827</td>
<td>75,456</td>
</tr>
<tr>
<td>INFRA-5</td>
<td>5</td>
<td>349</td>
<td>379</td>
<td>565</td>
</tr>
<tr>
<td>INFRA-3</td>
<td>3</td>
<td>15,126</td>
<td>29,861</td>
<td>28,023,500</td>
</tr>
</tbody>
</table>

• Abstract dependency structure

Fascinate – Experimental Results

• Effectiveness of dependency inference on BIO

---

Fascinate – Experimental Results

• Effectiveness of dependency inference on INFRA-5

<table>
<thead>
<tr>
<th>Methods</th>
<th>MAP</th>
<th>R-MPR</th>
<th>HLU</th>
<th>AUC</th>
<th>Prec@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fascinate</td>
<td>0.5040</td>
<td>0.3777</td>
<td>67.2231</td>
<td>0.8916</td>
<td>0.2500</td>
</tr>
<tr>
<td>Fascinate-Clust</td>
<td>0.4297</td>
<td>0.3220</td>
<td>56.8215</td>
<td>0.8159</td>
<td>0.2340</td>
</tr>
<tr>
<td>MulCol</td>
<td>0.4523</td>
<td>0.3239</td>
<td>59.8115</td>
<td>0.8329</td>
<td>0.2413</td>
</tr>
<tr>
<td>PairSid</td>
<td>0.3948</td>
<td>0.2392</td>
<td>49.5484</td>
<td>0.7413</td>
<td>0.2225</td>
</tr>
<tr>
<td>PairCol</td>
<td>0.3682</td>
<td>0.2489</td>
<td>48.5966</td>
<td>0.7406</td>
<td>0.2309</td>
</tr>
<tr>
<td>PairNMF</td>
<td>0.1315</td>
<td>0.0464</td>
<td>15.7148</td>
<td>0.5385</td>
<td>0.0711</td>
</tr>
<tr>
<td>PairRec</td>
<td>0.0970</td>
<td>0.0099</td>
<td>9.4853</td>
<td>0.5184</td>
<td>0.0399</td>
</tr>
<tr>
<td>FlatNMF</td>
<td>0.3212</td>
<td>0.2697</td>
<td>44.4654</td>
<td>0.7622</td>
<td>0.1999</td>
</tr>
<tr>
<td>FlatRec</td>
<td>0.1020</td>
<td>0.0778</td>
<td>11.5598</td>
<td>0.5740</td>
<td>0.0488</td>
</tr>
</tbody>
</table>

Overview of Part I

Part I: Recent Network Alignment (NA) Algorithms

- **Pairwise NA**
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes
    - w/ attributes
  - Optimal transport-based
    - w/o attributes

- **Collective NA**
  - Consistency-based
    - w/o attributes
    - w/ attributes
  - Embedding-based
    - w/o attributes
    - w/ attributes

- **Higher-Order NA**
  - Consistency-based
    - Single-level
    - Multilevel

- **Related Tasks**
  - Entity alignment
    - Non-GNN based
    - GNN-based
  - Cross-layer inference
    - Cross-network transformation
Cross-Network Node Associations

- Goal: to find node associations across different networks

Limitations of Traditional Methods

- Linear and/or consistency assumptions
  \[
  \min \| B_0 - PA_0P^T \|_F^2 \\
  \min \| \text{vec}(B_0) - \tilde{P} \text{vec}(A_0) \|_2^2
  \]
  Graph matching-based network alignment

- Embedding space disparity issue

\[
\min \| R - U_1^T U_2 \|_F^2 + \alpha \sum \text{Tr}(U_i^T(D_i - A_i)U_i)
\]
Factorization-based recommendation and cross-layer dependency inference

Cross-Network Transformation

- **Given:** (1) Source and target networks $G_1 = \{V_1, A_0, X_0\}$, $G_2 = \{V_2, B_0, Y_0\}$; Observed cross-network node associations $L$
- **Output:** (1) Cross-network transformation function $g$, s.t. $g(G_1) \approx G_2$; (2) Node association function $g_{node}$

NetTrans – Model Overview

- Key idea: encoder-decoder architecture
  - Encoder: to coarsen source network at different resolutions
  - Decoder: to reconstruct target network at different resolutions

NetTrans – Encoder

• Key component: TransPool as a pooling layer
• Supernode selection
  • Self-attention-based pooling

NetTrans – Encoder

- Supernode representation learning
  - Attention-based message passing
  - Aggregation by node-to-supernode assignment

NetTrans – Encoder

- **Node-to-supernode assignment**
  - Gumbel softmax to approximate $P$
  - Supernode candidate pruning

NetTrans – Encoder

• Supernode connections
• Use auxiliary connections $\widehat{A}_l$

$$A_l = \frac{1}{2} \left( A_{l-1}(I, I) + \widehat{A}_l \right)$$

**NetTrans – Decoder**

- **Goal:** to reconstruct target network
- **Key idea:** same latent meanings of supernodes
  - **Part #1:** leverage $\mathcal{G}_1$ by skip connections
  - **Part #2:** calibrate part #1 from supernodes to nodes

---

# NetTrans – Experimental Results

- Effectiveness of NetTrans for network alignment

<table>
<thead>
<tr>
<th></th>
<th>Cora1-Cora2</th>
<th></th>
<th>ACM-DBLP</th>
<th></th>
<th>Foursquare-Twitter</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy</td>
<td></td>
<td>Accuracy</td>
<td>Hits@10</td>
<td>Hits@30</td>
</tr>
<tr>
<td>NetTrans</td>
<td>90.98%</td>
<td>97.51%</td>
<td>89.89%</td>
<td>84.09%</td>
<td>94.52%</td>
<td>58.21%</td>
</tr>
<tr>
<td>FINAL-N</td>
<td>88.73%</td>
<td>90.77%</td>
<td>87.58%</td>
<td>82.91%</td>
<td>90.71%</td>
<td>54.39%</td>
</tr>
<tr>
<td>FINAL-P</td>
<td>62.28%</td>
<td>80.01%</td>
<td>54.34%</td>
<td>69.70%</td>
<td>83.12%</td>
<td>36.34%</td>
</tr>
<tr>
<td>REGAL</td>
<td>60.90%</td>
<td>69.20%</td>
<td>46.26%</td>
<td>63.68%</td>
<td>71.80%</td>
<td>41.78%</td>
</tr>
<tr>
<td>IONE</td>
<td>73.03%</td>
<td>79.92%</td>
<td>42.29%</td>
<td>58.93%</td>
<td>84.19%</td>
<td>33.00%</td>
</tr>
<tr>
<td>CrossMNA</td>
<td>59.06%</td>
<td>68.62%</td>
<td>33.26%</td>
<td>42.54%</td>
<td>49.69%</td>
<td>21.04%</td>
</tr>
</tbody>
</table>

**Observation:** NetTrans outperforms all other baselines for network alignment task

NetTrans – Experimental Results

• Effectiveness of NetTrans for social recommendation

<table>
<thead>
<tr>
<th></th>
<th>Ciao-0.2</th>
<th>Ciao-0.3</th>
<th>Ciao-0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prec@10</td>
<td>Rec@10</td>
<td>Rec@50</td>
<td>Prec@10</td>
</tr>
<tr>
<td>NetTrans</td>
<td>13.87%</td>
<td>11.08%</td>
<td>29.90%</td>
</tr>
<tr>
<td>BPR</td>
<td>1.37%</td>
<td>0.6%</td>
<td>20.25%</td>
</tr>
<tr>
<td>wpZAN</td>
<td>11.99%</td>
<td>9.19%</td>
<td>20.77%</td>
</tr>
<tr>
<td>GraphRec</td>
<td>8.65%</td>
<td>6.62%</td>
<td>17.56%</td>
</tr>
<tr>
<td>SamWalker</td>
<td>4.94%</td>
<td>1.97%</td>
<td>5.98%</td>
</tr>
<tr>
<td>NGCF</td>
<td>2.77%</td>
<td>1.21%</td>
<td>3.26%</td>
</tr>
</tbody>
</table>

**Observation:** NetTrans outperforms all other baselines for recommendation task

RoadMap

• Motivations and Background ✓
• Part I: Recent Network Alignment Algorithms ✓
• Part II: Network Alignment Applications
• Part III: Future Research Directions
Overview of Part II

Part II: Network Alignment Applications

Social Analysis
- User identity linkage
- Recommendation
  - Friends
  - Products
- Information diffusion

Bioinformatics
- Identify functional orthologs and knowledge transfer
  - Evolutionary relationships
  - Human aging
- Connectome Analysis

Knowledge Base
- Knowledge completion

Security
- Modeling adversarial activities
Social Analysis – User Identity Linkage

• User Identity Linkage
  • To identify the same physical user across social platforms

• Can be used for de-anonymization, information integration, etc.
User Identity Linkage

• Existing methods:
  • Profile based [Zafarani’13, Zhang’14, Perito’11, Vosecky’09]
  • Network based [Zhou’16, Zhang’15, Liu’16]
  • Profile + network based [Zhang’15, Shen’14, Zhang’16]

• Network-based can be considered as network alignment w/o attributes.

• Profile + network-based methods can be viewed as network alignment w/ attributes.
Social Analysis - Recommendation

• Friend recommendation:
  • For two social networks, if we know
    • User $u_1$ is a friend of user $u_2$ in $G_1$
    • User $v_1$ in $G_2$ and user $u_1$ in $G_1$ are same person
    • User $v_2$ in $G_2$ and user $u_2$ in $G_2$ are same person
    • But user $v_1$ and user $v_2$ are not friend in $G_2$
  • Then, we can recommend $v_1$ to user $v_2$

Cross-Site Friend Recommendation

- Think of it as a cross-site link prediction problem
- Given two incomplete social networks, we jointly solve network alignment and link prediction problems

![Diagram of network alignment and link prediction]

- : Observed edges
- : Imputed edges
- : Alignment
CENALP – Network Embedding

• DeepWalk-based network embedding
  • Key idea: build a world-view graph
    \[
    W = \begin{bmatrix}
    q \cdot P_{gg} & (1 - q) \cdot P_{gg'} \\
    (1 - q) \cdot P_{gg'} & q \cdot P_{gg'}
    \end{bmatrix}
    \]
  • Within-network node sampling with a probability of \(q\), and cross-network sampling with \((1 - q)\)
  • Allows for cross-network Skip-gram embedding

• Construction of \(P_{gg'}\) by structure and attribute
  \[
  \text{dist} = \left( \min_{d \in s_k(u)} \log(d + 1) - \min_{d \in s_k(u')} \log(d + 1) \right) + \left( \max_{d \in s_k(u)} \log(d + 1) - \max_{d \in s_k(u')} \log(d + 1) \right)
  \]
  \[
  \text{sim}_{\text{attr}}(u, u') = \frac{Y_u^\top \cdot Y_{u'}}{||Y_u||_2 \cdot ||Y_{u'}||_2},
  \]

CENALP – Network Alignment and Link Prediction

• Greedy alignment by embedding-based similarity
  • Given embeddings of \( u, u' \) in two networks
    \[
    \text{sim}_{\text{emb}}(u, u') = \frac{x_u^T \cdot x_{u'}}{\|x_u\|_2 \cdot \|x_{u'}\|_2}
    \]
  • Greedy-based alignment objective
    \[
    u^*, u'^* = \arg \max_{u, u'} \text{sim}_{\text{emb}}(u, u')
    \]

• Embedding for link prediction

---
CENALP – Algorithm

• Objective function

\[
\mathcal{L} = \sum_{\omega \in \text{walks}} \sum_{u_t \in \omega} \left[ \sum_{u_j \in C_{u_t}} \log \sigma (x_{u_j}^{\text{out}} \cdot x_{u_t}^{\text{in}}) 
+ \sum_{k=1}^{K_{\text{neg}}} \mathbb{E}_{u_k \sim R_k(u)} \log \sigma (-x_{u_k}^{\text{out}} \cdot x_{u_t}^{\text{in}}) \right].
\]

• Overall procedure

### CENALP – Experimental Results

- AUC score of link prediction

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>95%</th>
<th>90%</th>
<th>85%</th>
<th>80%</th>
<th>75%</th>
<th>70%</th>
<th>65%</th>
<th>60%</th>
<th>55%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBLP</td>
<td>JC [11]</td>
<td>92.28%</td>
<td>90.93%</td>
<td>90.37%</td>
<td>88.44%</td>
<td>85.43%</td>
<td>84.12%</td>
<td>80.51%</td>
<td>77.99%</td>
<td>74.81%</td>
<td>72.16%</td>
</tr>
<tr>
<td></td>
<td>AA [12]</td>
<td>92.33%</td>
<td>91.01%</td>
<td>90.30%</td>
<td>88.32%</td>
<td>85.44%</td>
<td>84.07%</td>
<td>80.48%</td>
<td>77.94%</td>
<td>74.82%</td>
<td>72.14%</td>
</tr>
<tr>
<td></td>
<td>SC [13]</td>
<td>78.84%</td>
<td>78.89%</td>
<td>78.68%</td>
<td>78.54%</td>
<td>74.72%</td>
<td>77.21%</td>
<td>76.83%</td>
<td>75.87%</td>
<td>73.91%</td>
<td>74.40%</td>
</tr>
<tr>
<td></td>
<td>n2v [22]</td>
<td>78.40%</td>
<td>81.34%</td>
<td>78.72%</td>
<td>78.32%</td>
<td>75.01%</td>
<td>74.30%</td>
<td>73.32%</td>
<td>71.60%</td>
<td>70.56%</td>
<td>68.61%</td>
</tr>
<tr>
<td>DBLP</td>
<td>n2v+LR</td>
<td>95.58%</td>
<td>95.44%</td>
<td>94.30%</td>
<td>94.30%</td>
<td>92.37%</td>
<td>93.07%</td>
<td>90.86%</td>
<td>90.80%</td>
<td>89.25%</td>
<td>88.08%</td>
</tr>
<tr>
<td>disturbed copy</td>
<td>CLF [8]</td>
<td>96.92%</td>
<td>96.55%</td>
<td>95.70%</td>
<td>95.69%</td>
<td>94.43%</td>
<td>94.28%</td>
<td>92.95%</td>
<td>92.93%</td>
<td>92.23%</td>
<td>90.68%</td>
</tr>
<tr>
<td></td>
<td>MNN [43]</td>
<td>98.13%</td>
<td>97.76%</td>
<td>97.57%</td>
<td>97.28%</td>
<td>97.10%</td>
<td>96.78%</td>
<td>96.38%</td>
<td>96.44%</td>
<td>96.03%</td>
<td>95.86%</td>
</tr>
<tr>
<td></td>
<td>CE-CLF</td>
<td>98.37%</td>
<td>98.29%</td>
<td>97.88%</td>
<td>97.80%</td>
<td>97.34%</td>
<td>96.74%</td>
<td>96.75%</td>
<td>96.56%</td>
<td>96.09%</td>
<td>95.86%</td>
</tr>
<tr>
<td></td>
<td>CELP</td>
<td>98.48%</td>
<td>98.39%</td>
<td>98.36%</td>
<td>98.57%</td>
<td>98.28%</td>
<td>98.49%</td>
<td>98.58%</td>
<td>98.31%</td>
<td>98.38%</td>
<td>98.27%</td>
</tr>
<tr>
<td></td>
<td>CENALP</td>
<td>99.65%</td>
<td>99.27%</td>
<td>99.35%</td>
<td>99.38%</td>
<td>99.36%</td>
<td>99.08%</td>
<td>99.01%</td>
<td>99.29%</td>
<td>99.25%</td>
<td>99.06%</td>
</tr>
<tr>
<td>Facebook</td>
<td>JC [11]</td>
<td>74.76%</td>
<td>77.89%</td>
<td>76.94%</td>
<td>75.64%</td>
<td>73.15%</td>
<td>72.63%</td>
<td>70.48%</td>
<td>68.90%</td>
<td>67.25%</td>
<td>65.12%</td>
</tr>
<tr>
<td></td>
<td>AA [12]</td>
<td>74.77%</td>
<td>77.54%</td>
<td>76.57%</td>
<td>75.76%</td>
<td>73.36%</td>
<td>72.68%</td>
<td>71.09%</td>
<td>68.84%</td>
<td>67.19%</td>
<td>65.18%</td>
</tr>
<tr>
<td></td>
<td>SC [13]</td>
<td>84.39%</td>
<td>83.39%</td>
<td>86.88%</td>
<td>84.53%</td>
<td>83.83%</td>
<td>83.79%</td>
<td>81.56%</td>
<td>80.80%</td>
<td>81.81%</td>
<td>77.61%</td>
</tr>
<tr>
<td>Twitter</td>
<td>n2v [22]</td>
<td>75.62%</td>
<td>78.94%</td>
<td>78.23%</td>
<td>79.36%</td>
<td>76.18%</td>
<td>75.25%</td>
<td>74.64%</td>
<td>74.86%</td>
<td>74.49%</td>
<td>71.16%</td>
</tr>
<tr>
<td></td>
<td>n2v+LR</td>
<td>82.20%</td>
<td>85.58%</td>
<td>83.04%</td>
<td>84.51%</td>
<td>81.64%</td>
<td>82.07%</td>
<td>81.17%</td>
<td>80.02%</td>
<td>78.08%</td>
<td>78.04%</td>
</tr>
<tr>
<td></td>
<td>CLF [8]</td>
<td>84.88%</td>
<td>85.02%</td>
<td>86.18%</td>
<td>86.70%</td>
<td>84.00%</td>
<td>83.99%</td>
<td>82.95%</td>
<td>82.43%</td>
<td>81.96%</td>
<td>80.75%</td>
</tr>
<tr>
<td></td>
<td>MNN [43]</td>
<td>95.72%</td>
<td>96.44%</td>
<td>96.28%</td>
<td>96.30%</td>
<td>96.21%</td>
<td>96.25%</td>
<td>96.07%</td>
<td>95.88%</td>
<td>95.47%</td>
<td>95.23%</td>
</tr>
<tr>
<td></td>
<td>CE-CLF</td>
<td>96.52%</td>
<td>96.84%</td>
<td>96.37%</td>
<td>96.34%</td>
<td>96.30%</td>
<td>95.69%</td>
<td>94.92%</td>
<td>94.31%</td>
<td>93.11%</td>
<td>92.00%</td>
</tr>
<tr>
<td></td>
<td>CELP</td>
<td>97.29%</td>
<td>97.52%</td>
<td>97.46%</td>
<td>97.85%</td>
<td>97.99%</td>
<td>97.56%</td>
<td>97.23%</td>
<td>97.15%</td>
<td>96.66%</td>
<td>96.14%</td>
</tr>
<tr>
<td></td>
<td>CENALP</td>
<td>97.31%</td>
<td>97.77%</td>
<td>97.51%</td>
<td>97.24%</td>
<td>97.60%</td>
<td>97.86%</td>
<td>97.34%</td>
<td>96.47%</td>
<td>96.74%</td>
<td>96.16%</td>
</tr>
</tbody>
</table>

Social Analysis - Recommendation

• Cross-site product recommendation:
  • Intuition: if users are aligned, purchase histories can be combined for better recommendation
  • Key idea: leverage cross-site actions to improve user modeling
  • Benefits: may mitigate issues, e.g., cold start, etc.

![Diagram showing cross-site actions and user modeling between Amazon and eBay.](image-url)
JUMA – Approach

• Key idea: use a probabilistic graphical model for joint user modeling over aligned sites

  • User’s site-specific preference $P_i^q$ is transferred from universal preference $U_i$ by transferring model $T^q$.

  • User conducts actions $A_i^q$ based on $P_i^q$ and site-specific item models $\{\phi_k^q\}$.

**JUMA – Approach**

• Joint user modeling over aligned sites

  • For item-based site (Douban), use matrix factorization method.

  • For text-based site (Weibo), use Latent Dirichlet Allocation (LDA) to model topic distributions for microblogs.

JUMA – Experimental Results

• Effectiveness of recommendation

<table>
<thead>
<tr>
<th>TARGET</th>
<th>ALGS</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text-Based</td>
<td>LDA</td>
<td>0.6514 ± 0.0017</td>
<td>0.6605 ± 0.0015</td>
<td>0.6694 ± 0.0016</td>
<td>0.6769 ± 0.0018</td>
<td>0.6839 ± 0.0015</td>
<td>0.6928 ± 0.0014</td>
</tr>
<tr>
<td></td>
<td>JUMA</td>
<td><strong>0.6824 ± 0.0014</strong></td>
<td><strong>0.6892 ± 0.0016</strong></td>
<td><strong>0.6976 ± 0.0014</strong></td>
<td><strong>0.7058 ± 0.0017</strong></td>
<td><strong>0.7120 ± 0.0012</strong></td>
<td><strong>0.7194 ± 0.0013</strong></td>
</tr>
<tr>
<td></td>
<td>CTR</td>
<td>0.7021 ± 0.0021</td>
<td>0.7133 ± 0.0017</td>
<td>0.7262 ± 0.0018</td>
<td>0.7352 ± 0.0017</td>
<td>0.7432 ± 0.0016</td>
<td>0.7532 ± 0.0015</td>
</tr>
<tr>
<td></td>
<td>JUMA</td>
<td><strong>0.7338 ± 0.0017</strong></td>
<td><strong>0.7420 ± 0.0015</strong></td>
<td><strong>0.7502 ± 0.0015</strong></td>
<td><strong>0.7592 ± 0.0015</strong></td>
<td><strong>0.7670 ± 0.0015</strong></td>
<td><strong>0.7743 ± 0.0014</strong></td>
</tr>
<tr>
<td>Item-Based</td>
<td>PMF</td>
<td>0.7275 ± 0.0016</td>
<td>0.7323 ± 0.0013</td>
<td>0.7384 ± 0.0015</td>
<td>0.7428 ± 0.0016</td>
<td>0.7485 ± 0.0014</td>
<td>0.7521 ± 0.0013</td>
</tr>
<tr>
<td></td>
<td>SVD++</td>
<td>0.7856 ± 0.0012</td>
<td>0.7929 ± 0.0010</td>
<td>0.7986 ± 0.0016</td>
<td>0.8055 ± 0.0012</td>
<td>0.8089 ± 0.0013</td>
<td>0.8112 ± 0.0011</td>
</tr>
<tr>
<td></td>
<td>TMF</td>
<td>0.7872 ± 0.0015</td>
<td>0.7946 ± 0.0012</td>
<td>0.8001 ± 0.0013</td>
<td>0.8071 ± 0.0019</td>
<td>0.8102 ± 0.0014</td>
<td>0.8132 ± 0.0012</td>
</tr>
<tr>
<td></td>
<td>mmTM</td>
<td>0.6929 ± 0.0019</td>
<td>0.6940 ± 0.0011</td>
<td>0.6943 ± 0.0012</td>
<td>0.6963 ± 0.0015</td>
<td>0.7034 ± 0.0010</td>
<td>0.7064 ± 0.0018</td>
</tr>
<tr>
<td></td>
<td>JUMA</td>
<td><strong>0.8127 ± 0.0017</strong></td>
<td><strong>0.8172 ± 0.0016</strong></td>
<td><strong>0.8219 ± 0.0013</strong></td>
<td><strong>0.8235 ± 0.0011</strong></td>
<td><strong>0.8243 ± 0.0015</strong></td>
<td><strong>0.8259 ± 0.0013</strong></td>
</tr>
</tbody>
</table>

**Observation:** JUMA performs best for both text-based site Weibo and item-based site Douban.

JUMA – Experimental Results

• Effectiveness of addressing cold-start

**Observation:** Improvements are higher when dealing with cold users than non-cold users.

Social Analysis – Information Diffusion

• Motivations
  • Users can post messages in multiple platforms;
  • Information thus propagates within-network and across networks.

![Diagram of two social networks, G1 and G2, with nodes A, B, C, D, E connected within each network and between the networks.](image-url)
M&M – Approach

• Goal: multi-aligned multi-relational network influence maximizer

• Key idea: to extend traditional linear threshold to depict diffusion across networks

• Activation probability functions:
  • For intra-network relation $i$
    \[ g_{v,i}^{(1)}(t+1) = \frac{\sum_{u \in \Gamma_{in}(v,i)} \phi^i_{(u,v)} \varphi(u,t)}{\sum_{u \in \Gamma_{in}(v,i)} \phi^i_{(u,v)}} \]
  • For inter-network relation $j$
    \[ h_{v,j}^{(1)}(t+1) = \frac{\sum_{u \in \Gamma_{in}(v,j)} \phi^j_{(u,v)} \varphi(u,t)}{\sum_{u \in \Gamma_{in}(v,j)} \phi^j_{(u,v)}} \]

M&M – Experimental Results

• Effectiveness of influence maximization
• Metric: # of activated users by the seed users

Overview of Part II

Part II: Network Alignment Applications

Social Analysis
- User identity linkage
- Recommendation
  - Friends
  - Products
- Information diffusion

Bioinformatics
- Identify functional orthologs and knowledge transfer
  - Evolutionary relationships
  - Human aging

Knowledge Base
- Knowledge completion

Security
- Modeling adversarial activities
Bioinformatics – Knowledge Transfer

• Motivations:
  • Traditional methods are based on sequence alignment
  • Network data and sequence data provide complementary insights
  • Restricting to sequences may limit knowledge transfer

• Network alignment to identify functional orthologs
  • Benefits: insightful for knowledge of aging and other biological processes.

Knowledge Transfer – Evolutionary Relationships Discovery

• Goal: using network alignment to guide biological knowledge transfer
  • From well-studied species to less well-studied species

• Methods:
  • GRAAL and H-GRAAL: focused on phylogenetic tree inference based on metabolic networks
  • MI-GRAAL:
    • Used these PPI network data to infer evolutionary relationships
    • Considered five herpesviruses based on their network similarities.

Knowledge Transfer – Human Aging Discovery

• Motivations:
  • Susceptibility to diseases increases with age
  • Important to study molecular mechanisms behind aging and aging-associated diseases

• Traditional methods:
  • Transferring knowledge from well-studied species to human between conserved sequence regions

• Network alignment-based methods:
  • MI-GRAAL and IsoRankN: align well known aging-related network parts of one species to known aging-related network parts of other species

Overview of Part II

Part II: Network Alignment Applications

Social Analysis
- User identity linkage
- Recommendation
  - Friends
  - Products
- Information diffusion

Bioinformatics
- Identify functional orthologs and knowledge transfer
  - Evolutionary relationships
  - Human aging

Knowledge Base
- Knowledge completion

Security
- Modeling adversarial activities

Applications
- Identify functional orthologs and knowledge transfer
  - Evolutionary relationships
  - Human aging

Knowledge completion
- Modeling adversarial activities
Knowledge Completion

• Goal: to complete a triple \((h, r, t)\) when one of \(h, r, t\) is missing

• Application scenario by entity alignment:
  • Two sets of triplets (i.e., KGs) for training
  • One set of triplets for testing
  • Two training KGs can be aligned

• Methods:
  • Basically can be any KG alignment methods
  • ITransE/IPTransE for example

ITransE – Experimental Results

- Effectiveness of ITransE for knowledge completion

<table>
<thead>
<tr>
<th>Metric</th>
<th>Mean Rank</th>
<th>Entity Prediction</th>
<th>Hits@10</th>
<th>Relation Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Raw</td>
<td>Filter</td>
<td>Raw</td>
<td>Filter</td>
</tr>
<tr>
<td>MTransE (LT)</td>
<td>240.8</td>
<td>131.3</td>
<td>36.4</td>
<td>47.3</td>
</tr>
<tr>
<td>MTransE (TB)</td>
<td>851.3</td>
<td>759.7</td>
<td>9.4</td>
<td>10.8</td>
</tr>
<tr>
<td>TransE</td>
<td>246.1</td>
<td>131.6</td>
<td>42.5</td>
<td>54.3</td>
</tr>
<tr>
<td>TransE + Aux</td>
<td>232.8</td>
<td>121.5</td>
<td>43.3</td>
<td>54.9</td>
</tr>
<tr>
<td>ITransE (SA)</td>
<td>209.2</td>
<td>101.0</td>
<td>44.2</td>
<td>55.1</td>
</tr>
<tr>
<td>PTransE</td>
<td>213.0</td>
<td>97.2</td>
<td>50.9</td>
<td>72.1</td>
</tr>
<tr>
<td>PTransE + Aux</td>
<td>206.3</td>
<td>80.4</td>
<td>52.7</td>
<td>80.7</td>
</tr>
<tr>
<td>IPTransE (SA)</td>
<td>197.5</td>
<td>70.6</td>
<td>53.0</td>
<td>80.8</td>
</tr>
</tbody>
</table>

**Observation:** By successfully leveraging the auxiliary information (i.e., second KG by alignment), ITransE and IPTransE perform better than other baseline methods.

Overview of Part II

Part II: Network Alignment Applications

Social Analysis
- User identity linkage
- Recommendation
  - Friends
  - Products
- Information diffusion

Bioinformatics
- Identify functional orthologs and knowledge transfer
  - Evolutionary relationships
  - Human aging

Knowledge Base
- Knowledge completion

Security
- Modeling adversarial activities
Security – Modeling Adversarial Activities

• Background:
  • Networks are natural structure to model adversarial activities
    • Smuggling
    • Illegal arm dealing
    • Illicit drug production
  • But such activities are often embedded in different domains

MAA – Challenges

• Domain heterogeneity
  • Communication networks
    • Phone call, emails, text, etc.
    • People who call each other may unlikely text often
    • Similarly, email network is structurally distinct from phone call network

• Spatial-temporal challenge
  • Relations contain much spatial-temporal information
    • Who calls whom at which location and at what time

• Very large-scale networks
MAA – Approaches

• Any scalable network alignment methods
  • w/o attribute: only based on connections
  • w/ attribute: view spatial-temporal information as attributes

• Encode temporal information:
  • Count # of connections in certain time window
  • Values at all time windows form node attributes
  • Can be used as attribute-based prior similarity matrix
  • And/or as the attributes in attributed alignment methods (e.g., FINAL)
RoadMap

• Motivations and Background ✔
• Part I: Recent Network Alignment Algorithms ✔
• Part II: Network Alignment Applications ✔
• Part III: Future Research Directions
Big Network Alignment – 4Vs

- 4V characteristics also hold for networks
Big Network Alignment – Volume

• Real-world networks are very large-scale
  • Facebook, Instagram, Twitter have billions of users

• **Challenge:** most of existing methods have at least $O(n^2)$ complexity
  • Some recent consistency-based and embedding-based methods reduce the complexity to linear
  • Complexity may be even larger if we handle multiple networks collectively

• **Question:** how to efficiently do network alignment?

• **Possible directions:** (1) leverage approximation techniques, (2) parallelizable algorithm
Big Network Alignment – Variety

• Real-world networks have rich information
  • Node/edge attributes, text descriptions, temporal information

• Methods exist to handle attribute information
  • But few can handle temporal relation information
  • Who called whom at what time, etc.

• **Question:** how to better incorporate side-information into network alignment?

• **Possible directions:** heterogeneous network alignment, temporal network alignment, etc.
Big Network Alignment – Variety

• Network heterogeneity
  • Networks to be aligned carry different types of information
  • Even same user may behave differently in different networks

• Existing methods explicitly or implicitly build upon consistency assumptions
  • But network heterogeneity may easily violate this assumption

• Questions:
  • How to align different types of networks (e.g., LinkedIn vs. FB)?
  • How to adaptively control consistency assumption?

• Possible directions: Deep learning methods that are highly learnable.
Big Network Alignment – Velocity

• Networks are dynamically changing over time.

• Dynamic network alignment
  • Simple solution: run from scratch at each timestamp
  • Limitation: time consuming; can’t capture dynamics

• Questions:
  • How to efficiently handle alignment over dynamic networks?
  • How to leverage the dynamics (e.g., smoothness)?

• Possible directions:
  • Matrix approximation to avoid unnecessary re-computations.
  • Dynamic network embedding-based alignment methods.
Big Network Alignment – Veracity

• Real-world networks are often noisy and incomplete.
  • Missing connections
  • Missing nodes
  • Missing attribute information

• Existing methods:
  • Jointly solve network alignment and link prediction
  • Benefit: if handled properly, they mutually benefit each other

• Challenge: error propagation
  • If alignment or imputed edges are not correct, the performance will be hurt.
Adversarial Network Alignment

- Improve the alignment effectiveness and robustness
- Noise/adversarial attacks can mislead alignment

 Rewiring attacks
Adversarial Network Alignment

• **Background:**
  • Existing adversarial attacks on network alignment are based on derivative-based importance score
  • But no work exits on adversarial defense

• **Challenge:**
  • Compared to adversarial attack/defense in single network, multiple networks may further complicate the defense process.

• **Possible direction:**
  • Graph neural network-based adversarial learning on network alignment
Integrated Network Alignment

• Explainable network alignment
  • **Background:** there exist explainable network mining tasks
    • Network embedding
    • Graph neural networks
    • Ranking, clustering, etc.
  • **Problem goal:**
    • Explain why two nodes should be aligned or not
  • **Possible directions:**
    • Extend explainable network embedding to embedding-based network alignment
Integrated Network Alignment

• Fair network alignment
  • Background:
    • Fairness has been studied recently in many machine learning and data mining tasks
    • Fairness in graphs has attracted attentions very recently, but for single network
  • Problem goal:
    • To debias the network alignment
  • Possible direction:
    • Extend fairness in single network mining to multiple networks first, then combine the specific objective of network alignment
Summary

• Background and motivation
  • Network alignment aims to find node correspondence across networks
  • A key step to many mining tasks across multiple networks
• Recent network alignment algorithms
  • Pairwise network alignment
  • Collective network alignment
  • Higher-order network alignment
  • Other related tasks
• Network alignment applications
• Future directions
References


References


References


