AHP: Learning to Negative Sample for Hyperedge Prediction

Hyunjin Hwang*
Seungwoo Lee*
Chanyoung Park
Kijung Shin
Backgrounds (1) Hypergraphs

• **Hypergraphs,** $H(V, E)$
  - Hypergraphs represent interactions among multiple nodes
  - Each hyperedge $e \in E$ is a subset of node set $V$

• **Advantages:**
  - Have more expressive power than ordinary graphs
  - Capture higher-order information

![Diagram of hypergraphs and papers with authors](image)
Backgrounds (2) Hyperedge Prediction

• **Hyperedge prediction**: Predict future or unobserved hyperedges

• **Applications**
  • Collaboration recommendation
  • Chemical reaction prediction
  • Recipe discovery / Drug discovery
  • E-mail recipient recommendation
Problem Definition

• **True hyperedge set** $E \cup E'$
  - $E$: observed hyperedge set
  - $E'$: unobserved hyperedge set (target set)

• **Hyperedge candidate, $c$**
  - A subset of nodes that can be a hyperedge, but currently not observed ($c \notin E$)

• **Problem definition. (Hyperedge prediction)**
  - **Given**: a hypergraph $G(V, E)$ and a hyperedge candidate $c$
  - **Classify**: candidate $c$ whether it would be in the target set $E'$ or not
Challenges and Motivation

• Vast sample space
  • The number of possible hyperedge candidates is $2^{|V|}$
  • Sampling hyperedge candidates for classification task is inevitable

• Poor generalization ability
  • NN models should generalize on the sampled hyperedge candidates to all
  • Heuristic sampling schemes in previous works generalize poorly

All subsets of 15,639 nodes (from DBLP)
Challenges and Motivation

• Vast sample space
  • The number of possible hyperedge candidates is $2^{|V|}$
  • Sampling hyperedge candidates for classification task is inevitable

• Poor generalization ability
  • NN models should generalize on the sampled hyperedge candidates to all
  • Heuristic sampling schemes in previous works generalize poorly

All subsets of 15,639 nodes (from DBLP)

- about $10^{4707}$ hyperedge candidates
- $2^V \setminus E$
- about $2.3 \times 10^4$ observed hyperedges
- $E'$
- unobserved hyperedges
Previous Approaches

**Positive examples** (hyperedges)

Sampled negative examples

**Negative example sampler**

(1) sample

**HyperGNN model**

(2) classify

HNHN → POOL → MLP

Node embeddings → Candidate embeddings → Candidate scores

Classification loss

(3) update
Proposed: Adversarial Hyperedge Prediction (AHP)

- **Positive examples (hyperedges)**
- **Generated negative examples**

**Generator**
- Sample $k$
- Top $k$
- Random noise
- Binary vector for node membership
- MLP

**Discriminator**
- HNHN
- POOL
- MLP
- Candidate scores

(1) sample
(2) classify
(3) update
Experiments (1) Datasets

- **Collaboration dataset** (node: author, hyperedge: authors in a paper)
- **Authorship dataset** (node: paper, hyperedge: papers by an author)
- **Co-citation dataset** (node: paper, edge: cited papers in a paper)

<table>
<thead>
<tr>
<th>Property</th>
<th>Citeseer</th>
<th>Cora</th>
<th>Cora-A</th>
<th>Pubmed</th>
<th>DBLP-A</th>
<th>DBLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Co-citation</td>
<td>Co-citation</td>
<td>Authorship</td>
<td>Co-citation</td>
<td>Authorship</td>
<td>Collaboration</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>1,457</td>
<td>1,434</td>
<td>2,388</td>
<td>3,840</td>
<td>39,283</td>
<td>15,639</td>
</tr>
<tr>
<td>Number of edges</td>
<td>1,078</td>
<td>1,579</td>
<td>1,072</td>
<td>7,962</td>
<td>16,483</td>
<td>22,964</td>
</tr>
<tr>
<td>Average size of hyperedges</td>
<td>3.2</td>
<td>3</td>
<td>4.3</td>
<td>4.3</td>
<td>4.5</td>
<td>2.7</td>
</tr>
<tr>
<td>Maximum size of hyperedges</td>
<td>26</td>
<td>5</td>
<td>43</td>
<td>171</td>
<td>80</td>
<td>18</td>
</tr>
<tr>
<td>Minimum size of hyperedges</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Dimension of node feature</td>
<td>3,703</td>
<td>1,433</td>
<td>1,433</td>
<td>500</td>
<td>4543</td>
<td>4543</td>
</tr>
</tbody>
</table>
## Experiments (2) Baselines

<table>
<thead>
<tr>
<th>Model</th>
<th>Node embedding</th>
<th>Edge pooling</th>
<th>Negative sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>NHP</td>
<td>NHP</td>
<td>Maxmin</td>
<td>Half corruption</td>
</tr>
<tr>
<td>HyperSAGNN</td>
<td>Random walk</td>
<td>HyperSAGNN</td>
<td>MNS</td>
</tr>
<tr>
<td>N-order expansion</td>
<td>Six features from given candidate</td>
<td>Stars in clique expansion</td>
<td></td>
</tr>
<tr>
<td>AHP-S</td>
<td>HNHN</td>
<td>Maxmin</td>
<td>SNS</td>
</tr>
<tr>
<td>AHP-M</td>
<td>HNHN</td>
<td>Maxmin</td>
<td>MNS</td>
</tr>
<tr>
<td>AHP-C</td>
<td>HNHN</td>
<td>Maxmin</td>
<td>CNS</td>
</tr>
<tr>
<td>AHP-S+M+C</td>
<td>HNHN</td>
<td>Maxmin</td>
<td>SNS+MNS+CNS</td>
</tr>
<tr>
<td>AHP</td>
<td>HNHN</td>
<td>Maxmin</td>
<td>Generator</td>
</tr>
</tbody>
</table>
Experiments (3) Results: Generalization Ability

• State of the art methods have **poor generalization ability**
Experiments (3) Results: Overall Performance

• Our proposed method, AHP, outperforms on average

<table>
<thead>
<tr>
<th>Predictive Performance</th>
<th>Cora-A</th>
<th>Cora</th>
<th>Citeseer</th>
<th>Pubmed</th>
<th>DBLP</th>
<th>DBLP-A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUROC</td>
<td>AP</td>
<td>AUROC</td>
<td>AP</td>
<td>AUROC</td>
<td>AP</td>
</tr>
<tr>
<td>Expansion</td>
<td>0.656</td>
<td>0.706</td>
<td>0.477</td>
<td>0.607</td>
<td>0.591</td>
<td>0.681</td>
</tr>
<tr>
<td>NHP</td>
<td>0.723</td>
<td>0.748</td>
<td>0.703</td>
<td>0.718</td>
<td>0.751</td>
<td>0.751</td>
</tr>
<tr>
<td>HyperSAGNN</td>
<td>0.506</td>
<td>0.571</td>
<td>0.545</td>
<td>0.584</td>
<td>0.475</td>
<td>0.521</td>
</tr>
<tr>
<td>AHP-S</td>
<td>0.872</td>
<td>0.869</td>
<td>0.777</td>
<td>0.764</td>
<td>0.781</td>
<td>0.787</td>
</tr>
<tr>
<td>AHP-M</td>
<td>0.873</td>
<td>0.874</td>
<td>0.777</td>
<td>0.758</td>
<td>0.748</td>
<td>0.756</td>
</tr>
<tr>
<td>AHP-C</td>
<td>0.876</td>
<td>0.878</td>
<td>0.767</td>
<td>0.749</td>
<td>0.777</td>
<td>0.786</td>
</tr>
<tr>
<td>AHP-S+M+C</td>
<td>0.874</td>
<td>0.878</td>
<td>0.771</td>
<td>0.753</td>
<td>0.774</td>
<td>0.779</td>
</tr>
<tr>
<td>AHP (Proposed)</td>
<td><strong>0.888</strong></td>
<td><strong>0.882</strong></td>
<td><strong>0.799</strong></td>
<td><strong>0.772</strong></td>
<td><strong>0.824</strong></td>
<td><strong>0.819</strong></td>
</tr>
</tbody>
</table>
Conclusions

• Observation: **Heuristic sampling schemes limit the generalization ability** of deep learning-based hyperedge-prediction.

• Algorithm: **AHP learns to sample negative examples** by adversarial training for better generalization

• Results: In terms of AUROC, **AHP is up to 28.2% better than best existing methods** and up to 5.5% better than variants with sampling schemes tailored to test sets

Code & Datasets: [https://github.com/HyunjinHwn/SIGIR22-AHP](https://github.com/HyunjinHwn/SIGIR22-AHP)
Thank you!