Neural Graph Collaborative Filtering

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, Tat-Seng Chua

22 July 2019
Information Isolated Island Effect:
- Model each instance individually
- While overlooking relations among instances
- Might result in suboptimal performance:
  - Making an instance's representation dependent only on its own features
  - Making interactions suffer from sparse issues

Limited Representation Ability
Suboptimal Model Capacity
One Solution: Reorganizing Data into Graphs!

The data is more **closely connected** than we might think!

**User-Item Interactions**

**User/Item Profiles**

**Knowledge Graph**

**Limited Representation Ability** → **Information Propagation along with the connections**

**Suboptimal Model Capacity** → **High-order connectivity complementary to user-item interactions**
• **Collaborative Filtering (CF)** is the most well-known technique for recommendation.
  
  • Homophily assumption: a user preference can be predicted from his/her similar users.

• **Collaborative Signals -> Behavioral Similarity of users**
  
  • if $u_1$ and $u_3$ have interacted with the same items \{$i_1, i_3$\}, $u_1$ is likely to have similar preferences on other items \{$i_4$\}. 

```
\[ u_1 \text{ Alice} \quad u_2 \text{ Annie} \quad u_3 \text{ Bob} \]
\[ i_1 \text{ Shape of You} \quad i_2 \text{ I See Fire} \quad i_3 \text{ Castle on the Hill} \quad i_4 \text{ Skin} \quad i_5 \text{ Lose Yourself} \]
```
Revisiting CF via High-order Connectivity

High-order Connectivity from User-item Bipartite Graphs

Why $u_1$ may like $i_4$?

Existing CF methods (e.g., MF, FISM, AutoRec) don't model high-order connectivity explicitly.
- Embedding function only considers descriptive features (e.g., ID, attributes)
- User-item interactions are not considered

Our contribution: CF modeling with high-order connectivity via GNN.

• Definition: the paths that reach $u_1$ from any node with the path length $l$ larger than 1.

• A natural way to encode collaborative signals
Embedding Propagation, inspired by GNNs

- Propagate embeddings recursively on the graph
- Construct information flows in the embedding space

**First-order Propagation**

- **Message Construction**: generate message from one neighbor

  \[ m_{u \leftarrow i} = \frac{1}{\sqrt{|N_u||N_i|}} \left( W_1 e_i + W_2 (e_i \odot e_u) \right) \]

- **Message Aggregation**: update ego node’s representation by aggregating message from all neighbors

  \[ e^{(1)}_u = \text{LeakyReLU} \left( m_{u \leftarrow u} + \sum_{i \in N_u} m_{u \leftarrow i} \right) \]

- message dependent on the affinity, distinct from GCN, GraphSage, etc.
- Pass more information to similar nodes
- Self-connections: all neighbors of \( u \)
**High-order Propagation**

- We stack more embedding propagation layers to explore the high-order connectivity information.

\[
e^{(l)}_u = \text{LeakyReLU}(m^{(l)}_{u\leftarrow u} + \sum_{i \in \mathcal{N}_u} m^{(l)}_{u\leftarrow i}),
\]

representation of \( u \) at the \( l \)-th layer

- The collaborative signal like \( u_1 \leftarrow i_2 \leftarrow u_2 \leftarrow i_4 \) can be captured in the embedding propagation process.

- Collaborative signal can be injected into the representation learning process.
The representations at different layers
• emphasize the messages passed over different connections
• have different contributions in reflecting user preference
## Experiments

### Datasets
- Gowalla, Amazon-Book, Yelp2018

### Evaluation Metrics
- recall@K, ndcg@K

### Baselines

<table>
<thead>
<tr>
<th>Data for Embedding Function</th>
<th>Connectivity</th>
<th>Aggregation Type in GNNs</th>
<th>Jump Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF ID</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NeuFM ID</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CMN Personal History</td>
<td>First-order</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HOP-Rec Multi-hop Neighbors</td>
<td>High-order</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PinSage Collaborative Signals</td>
<td>Second-order</td>
<td>Concatenation</td>
<td>-</td>
</tr>
<tr>
<td>GC-MC Collaborative Signals</td>
<td>First-order</td>
<td>Sum</td>
<td>-</td>
</tr>
<tr>
<td>NGCF Collaborative Signals</td>
<td><strong>High-order</strong></td>
<td><strong>Sum + Element-wise Product</strong></td>
<td><strong>Jump Knowledge</strong></td>
</tr>
</tbody>
</table>
• NGCF consistently yields the best performance on all the datasets.

• This verifies the importance of capturing collaborative signal in embedding function.
NGCF and HOP-Rec consistently outperform all other baselines on most user groups.

Exploiting high-order connectivity facilitates the representation learning for inactive users.

It might be promising to solve the sparsity issue in recommender systems.
Increasing the depth of NGCF substantially enhances the recommendation cases.

User similarity and collaborative signal are carried by the second- and third-order connectivity, respectively.
Effect of High-order Connectivity (2)

- The points with the same colors (i.e., the items consumed by the same users) tend to form the clusters.

- The **connectivities of users and items are well reflected in the embedding space**, that is, they are embedded into the near part of the space.
Take-home messages

• Modeling high-order connectivity from user-item interactions is important for CF.
• We proposed a GNN model to do this in an end-to-end way.

Future Work

• Incorporating knowledge graph into NGCF [Wang et al. KDD’19]
• Automating NGCF, e.g., negative sampling, hyper-parameters
• Keep Human-in-the-loop: conversational recommendation
THANK YOU!

Neural Graph Collaborative Filtering, SIGIR2019;
http://staff.ustc.edu.cn/~hexn/papers/sigir19-NGCF.pdf