Improving Recommendation via Contrastive Learning

Yupeng Hou
04/13/2022
Outline

• [5 min] Background - Contrastive Learning
• [10 min] Improving Collaborative Filtering via CL
• [8 min] Improving Session-based RecSys via CL
• [2 min] Conducting Research with RecBole
• [5 min] QA
Background

Contrastive Learning
Background - Contrastive Learning

Which pair is more similar 😏

1 ⬤

2 ⬤

Background - Contrastive Learning

• Arts of sample pairs and similarity

\[ \text{score}(f(x), f(x^+)) \gg \text{score}(f(x), f(x^-)) \]

• Target: better encoder \( f \)
  • In other words, better distribution of encoded representations

Background - Contrastive Learning

- Arts of sample pairs and similarity (more mathematically)

\[ \mathcal{L}_N = - \mathbb{E}_X \left[ \log \frac{\exp(f(x)^T f(x^+))}{\exp(f(x)^T f(x^+)) + \sum_{j=1}^{N-1} \exp(f(x)^T f(x_j))} \right] \]
Background - Contrastive Learning

• Key points of CL
  • (Very) similar and easily-obtained positive pairs (mostly self-supervised, via data augmentation);
  • (Very) large amount of negative pairs;
Background - Contrastive Learning

• Examples for self-supervised signal in CV

Background - Contrastive Learning

• Examples for self-supervised signal for graphs

Wu et al. Self-supervised Graph Learning for Recommendation. SIGIR 2021.
Background - Contrastive Learning

• Examples for self-supervised signal via dropout

Background - Contrastive Learning

• Examples for self-supervised signal in recommendation?
  • Directly transfer existing methods 🙆
  • Self-supervised signals specifically for RecSys 🤔
    • easily obtained;
    • valuable;
Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning.

Background - Graph Collaborative Filtering

(General Recommendation)

User-Item Interaction Graph

High-order Connectivity for $u_1$

$e^{(k+1)}_i = \sum_{u \in |N_i|} \frac{1}{\sqrt{|N_i|}} e^{(k)}_i$

$e^{(k+1)}_u = \sum_{i \in |N_u|} \frac{1}{\sqrt{|N_u|}} e^{(k)}_i$

LightGCN

Wang et al. Neural Graph Collaborative Filtering. SIGIR 2019.
Challenge

- U-I graph is usually **sparse or noisy**;

- Lack of explicitly modeling **high-order constraints** on U-I graph; (e.g. U-U, or I-I)
Idea

- high-order constraints (neighbors) as CL supervision signal

- -> alleviating sparsity and noisy issue;
Neighborhood-enriched CL

• Structural Neighbors

• Semantic Neighbors
Structural Neighbors

• Structurally connected nodes by high-order paths

$$\mathcal{L}_{struct} = - \log \frac{\exp(\text{o})}{\sum \exp(\text{o})}$$
Structural Neighbors

• How to **efficiently** obtain high-order neighbors' representations?
Structural Neighbors

• How to **efficiently** obtain high-order neighbors' representations?
Structural Neighbors

• How to efficiently obtain high-order neighbors' representations?

\[
\mathcal{L}_{struc} = - \log \frac{\exp(\text{○○})}{\sum \exp(\text{○○○○})}
\]

\[
\mathcal{L}^U = \sum_{u \in U} - \log \frac{\exp((z_u^{(k)} \cdot z_u^{(0)}) / \tau))}{\sum_{v \in U} \exp((z_u^{(k)} \cdot z_v^{(0)}) / \tau))},
\]

\[
\mathcal{L}_S = \mathcal{L}^U + \alpha \mathcal{L}^I.
\]
Semantic Neighbors

- Semantically similar neighbors which may not be directly reachable on graphs

\[
\mathcal{L}_{\text{proto}} = -\log \frac{\exp(\circ \triangle)}{\exp(\circ \triangle) + \exp(\circ \triangle) + \ldots}
\]

Neighbors in embedding space
Semantic Neighbors

• Semantically similar neighbors which may not be directly reachable on graphs

\[ L_{\text{proto}} = -\log \frac{\exp(\bigcirc \triangle)}{\exp(\bigcirc \triangle) + \exp(\bigcirc \triangle) + \ldots} \]

\[ L_P^U = \sum_{u \in U} -\log \frac{\exp(e_u \cdot c_i / \tau)}{\sum_{c_j \in C} \exp(e_u \cdot c_j / \tau)} \]

\[ L_P = L_P^U + \alpha L_P^I. \]

K-means
(optimized via EM)
NCL overall

\[ \mathcal{L} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_S + \lambda_2 \mathcal{L}_P + \lambda_3 \| \Theta \|_2, \]

arbitrary GCF algorithm

NCL
NCL experiments

• 5 widely-used public datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>#Users</th>
<th>#Items</th>
<th>#Interactions</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-1M</td>
<td>6,040</td>
<td>3,629</td>
<td>836,478</td>
<td>0.03816</td>
</tr>
<tr>
<td>Yelp</td>
<td>45,478</td>
<td>30,709</td>
<td>1,777,765</td>
<td>0.00127</td>
</tr>
<tr>
<td>Books</td>
<td>58,145</td>
<td>58,052</td>
<td>2,517,437</td>
<td>0.00075</td>
</tr>
<tr>
<td>Gowalla</td>
<td>29,859</td>
<td>40,989</td>
<td>1,027,464</td>
<td>0.00084</td>
</tr>
<tr>
<td>Alibaba</td>
<td>300,000</td>
<td>81,614</td>
<td>1,607,813</td>
<td>0.00007</td>
</tr>
</tbody>
</table>

• Carefully hyper-parameter tuning for all baselines

• [https://github.com/RUCAIBox/NCL](https://github.com/RUCAIBox/NCL)
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>BPRMF</th>
<th>NeuMF</th>
<th>FISM</th>
<th>NGCF</th>
<th>MultiGCCF</th>
<th>DGCF</th>
<th>LightGCN</th>
<th>SGL</th>
<th>NCL</th>
<th>Improv.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MovieLens-1M</strong></td>
<td>Recall@10</td>
<td>0.1804</td>
<td>0.1657</td>
<td>0.1887</td>
<td>0.1846</td>
<td>0.1830</td>
<td>0.1811</td>
<td>0.1876</td>
<td>0.1888</td>
<td><strong>0.2057</strong> *</td>
<td>+8.95%</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.2463</td>
<td>0.2295</td>
<td>0.2494</td>
<td>0.2528</td>
<td>0.2510</td>
<td>0.2520</td>
<td>0.2514</td>
<td>0.2526</td>
<td><strong>0.2732</strong> *</td>
<td>+8.07%</td>
</tr>
<tr>
<td></td>
<td>Recall@20</td>
<td>0.2714</td>
<td>0.2520</td>
<td>0.2798</td>
<td>0.2741</td>
<td>0.2759</td>
<td>0.2779</td>
<td>0.2796</td>
<td>0.2848</td>
<td><strong>0.3037</strong> *</td>
<td>+6.63%</td>
</tr>
<tr>
<td></td>
<td>NDCG@20</td>
<td>0.2569</td>
<td>0.2400</td>
<td>0.2607</td>
<td>0.2614</td>
<td>0.2617</td>
<td>0.2615</td>
<td>0.2620</td>
<td>0.2649</td>
<td><strong>0.2843</strong> *</td>
<td>+7.32%</td>
</tr>
<tr>
<td></td>
<td>Recall@50</td>
<td>0.4300</td>
<td>0.4122</td>
<td>0.4421</td>
<td>0.4341</td>
<td>0.4364</td>
<td>0.4424</td>
<td>0.4469</td>
<td>0.4487</td>
<td><strong>0.4686</strong> *</td>
<td>+4.44%</td>
</tr>
<tr>
<td></td>
<td>NDCG@50</td>
<td>0.3014</td>
<td>0.2851</td>
<td>0.3078</td>
<td>0.3055</td>
<td>0.3056</td>
<td>0.3078</td>
<td>0.3091</td>
<td>0.3111</td>
<td><strong>0.3300</strong> *</td>
<td>+6.08%</td>
</tr>
<tr>
<td><strong>Yelp</strong></td>
<td>Recall@10</td>
<td>0.0643</td>
<td>0.0531</td>
<td>0.0714</td>
<td>0.0630</td>
<td>0.0646</td>
<td>0.0723</td>
<td>0.0730</td>
<td>0.0833</td>
<td><strong>0.0920</strong> *</td>
<td>+10.44%</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.0458</td>
<td>0.0377</td>
<td>0.0510</td>
<td>0.0446</td>
<td>0.0450</td>
<td>0.0514</td>
<td>0.0520</td>
<td>0.0601</td>
<td><strong>0.0678</strong> *</td>
<td>+12.81%</td>
</tr>
<tr>
<td></td>
<td>Recall@20</td>
<td>0.1043</td>
<td>0.0885</td>
<td>0.1119</td>
<td>0.1026</td>
<td>0.1053</td>
<td>0.1135</td>
<td>0.1163</td>
<td>0.1288</td>
<td><strong>0.1377</strong> *</td>
<td>+6.91%</td>
</tr>
<tr>
<td></td>
<td>NDCG@20</td>
<td>0.0580</td>
<td>0.0486</td>
<td>0.0636</td>
<td>0.0567</td>
<td>0.0575</td>
<td>0.0641</td>
<td>0.0652</td>
<td>0.0739</td>
<td><strong>0.0817</strong> *</td>
<td>+10.55%</td>
</tr>
<tr>
<td></td>
<td>Recall@50</td>
<td>0.1862</td>
<td>0.1654</td>
<td>0.1963</td>
<td>0.1864</td>
<td>0.1882</td>
<td>0.1989</td>
<td>0.2016</td>
<td>0.2140</td>
<td><strong>0.2247</strong> *</td>
<td>+5.00%</td>
</tr>
<tr>
<td></td>
<td>NDCG@50</td>
<td>0.0793</td>
<td>0.0685</td>
<td>0.0856</td>
<td>0.0784</td>
<td>0.0790</td>
<td>0.0862</td>
<td>0.0875</td>
<td>0.0964</td>
<td><strong>0.1046</strong> *</td>
<td>+8.51%</td>
</tr>
<tr>
<td><strong>Amazon-Books</strong></td>
<td>Recall@10</td>
<td>0.0607</td>
<td>0.0507</td>
<td>0.0721</td>
<td>0.0617</td>
<td>0.0625</td>
<td>0.0737</td>
<td>0.0797</td>
<td>0.0898</td>
<td><strong>0.0933</strong> *</td>
<td>+3.90%</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.0431</td>
<td>0.0351</td>
<td>0.0504</td>
<td>0.0427</td>
<td>0.0433</td>
<td>0.0521</td>
<td>0.0565</td>
<td>0.0645</td>
<td><strong>0.0679</strong> *</td>
<td>+5.27%</td>
</tr>
<tr>
<td></td>
<td>Recall@20</td>
<td>0.0956</td>
<td>0.0823</td>
<td>0.1099</td>
<td>0.0978</td>
<td>0.0991</td>
<td>0.1128</td>
<td>0.1206</td>
<td>0.1331</td>
<td><strong>0.1381</strong> *</td>
<td>+3.76%</td>
</tr>
<tr>
<td></td>
<td>NDCG@20</td>
<td>0.0537</td>
<td>0.0447</td>
<td>0.0622</td>
<td>0.0537</td>
<td>0.0545</td>
<td>0.064</td>
<td>0.0689</td>
<td>0.0777</td>
<td><strong>0.0815</strong> *</td>
<td>+4.89%</td>
</tr>
<tr>
<td></td>
<td>Recall@50</td>
<td>0.1681</td>
<td>0.1447</td>
<td>0.183</td>
<td>0.1699</td>
<td>0.1688</td>
<td>0.1908</td>
<td>0.2012</td>
<td>0.2157</td>
<td><strong>0.2175</strong> *</td>
<td>+0.83%</td>
</tr>
<tr>
<td></td>
<td>NDCG@50</td>
<td>0.0726</td>
<td>0.061</td>
<td>0.0815</td>
<td>0.0725</td>
<td>0.0727</td>
<td>0.0843</td>
<td>0.0899</td>
<td>0.0992</td>
<td><strong>0.1024</strong> *</td>
<td>+3.23%</td>
</tr>
<tr>
<td><strong>Gowalla</strong></td>
<td>Recall@10</td>
<td>0.1158</td>
<td>0.1039</td>
<td>0.1081</td>
<td>0.1192</td>
<td>0.1108</td>
<td>0.1252</td>
<td>0.1362</td>
<td>0.1465</td>
<td><strong>0.1500</strong> *</td>
<td>+2.39%</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.0833</td>
<td>0.0731</td>
<td>0.0755</td>
<td>0.0852</td>
<td>0.0791</td>
<td>0.0902</td>
<td>0.0876</td>
<td>0.1048</td>
<td><strong>0.1082</strong> *</td>
<td>+3.24%</td>
</tr>
<tr>
<td></td>
<td>Recall@20</td>
<td>0.1695</td>
<td>0.1535</td>
<td>0.1620</td>
<td>0.1755</td>
<td>0.1626</td>
<td>0.1829</td>
<td>0.1976</td>
<td>0.2084</td>
<td><strong>0.2133</strong> *</td>
<td>+2.35%</td>
</tr>
<tr>
<td></td>
<td>NDCG@20</td>
<td>0.0988</td>
<td>0.0873</td>
<td>0.0913</td>
<td>0.1013</td>
<td>0.0940</td>
<td>0.1066</td>
<td>0.1152</td>
<td>0.1225</td>
<td><strong>0.1265</strong> *</td>
<td>+3.27%</td>
</tr>
<tr>
<td></td>
<td>Recall@50</td>
<td>0.2756</td>
<td>0.2510</td>
<td>0.2673</td>
<td>0.2811</td>
<td>0.2631</td>
<td>0.2877</td>
<td>0.3044</td>
<td>0.3197</td>
<td><strong>0.3259</strong> *</td>
<td>+1.94%</td>
</tr>
<tr>
<td></td>
<td>NDCG@50</td>
<td>0.1450</td>
<td>0.1110</td>
<td>0.1169</td>
<td>0.1270</td>
<td>0.1184</td>
<td>0.1322</td>
<td>0.1414</td>
<td>0.1497</td>
<td><strong>0.1542</strong> *</td>
<td>+3.01%</td>
</tr>
<tr>
<td><strong>Alibaba-iFashion</strong></td>
<td>Recall@10</td>
<td>0.303</td>
<td>0.182</td>
<td>0.3075</td>
<td>0.0832</td>
<td>0.0401</td>
<td>0.0447</td>
<td>0.0457</td>
<td>0.0461</td>
<td><strong>0.0477</strong> *</td>
<td>+3.47%</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.0161</td>
<td>0.0092</td>
<td>0.0190</td>
<td>0.0198</td>
<td>0.0207</td>
<td>0.0241</td>
<td>0.0246</td>
<td>0.0248</td>
<td><strong>0.0259</strong> *</td>
<td>+4.44%</td>
</tr>
<tr>
<td></td>
<td>Recall@20</td>
<td>0.0467</td>
<td>0.0302</td>
<td>0.0553</td>
<td>0.0615</td>
<td>0.0634</td>
<td>0.0677</td>
<td>0.0692</td>
<td>0.0692</td>
<td><strong>0.0713</strong> *</td>
<td>+3.03%</td>
</tr>
<tr>
<td></td>
<td>NDCG@20</td>
<td>0.0203</td>
<td>0.0123</td>
<td>0.0239</td>
<td>0.0257</td>
<td>0.0266</td>
<td>0.0299</td>
<td>0.0246</td>
<td>0.0307</td>
<td><strong>0.0319</strong> *</td>
<td>+3.01%</td>
</tr>
<tr>
<td></td>
<td>Recall@50</td>
<td>0.0799</td>
<td>0.0576</td>
<td>0.0943</td>
<td>0.1081</td>
<td>0.1107</td>
<td>0.1120</td>
<td>0.1144</td>
<td>0.1141</td>
<td><strong>0.1165</strong> *</td>
<td>+1.84%</td>
</tr>
<tr>
<td></td>
<td>NDCG@50</td>
<td>0.0269</td>
<td>0.0177</td>
<td>0.0317</td>
<td>0.0349</td>
<td>0.0360</td>
<td>0.0387</td>
<td>0.0396</td>
<td>0.0396</td>
<td><strong>0.0409</strong> *</td>
<td>+3.28%</td>
</tr>
</tbody>
</table>
NCL experiments (2)

• Ablation Study
NCL experiments (3)

• Impact of Data Sparsity Levels

Figure 4: Performance analysis for different sparsity-level users (Recall@10). G1 denotes the group of users with the lowest average number of interactions.
NCL experiments (4)

- Effect of Structural Neighbors

Table 3: Performance comparison w.r.t. different hop of structural neighbors.

<table>
<thead>
<tr>
<th>Hop</th>
<th>MovieLens-1M Recall@10</th>
<th>NDCG@10</th>
<th>Yelp Recall@10</th>
<th>NDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o s-n</td>
<td>0.1876</td>
<td>0.2514</td>
<td>0.0730</td>
<td>0.0520</td>
</tr>
<tr>
<td>1</td>
<td>0.2057</td>
<td>0.2732</td>
<td>0.0920</td>
<td>0.0678</td>
</tr>
<tr>
<td>2</td>
<td>0.1838</td>
<td>0.2516</td>
<td>0.0837</td>
<td>0.0602</td>
</tr>
<tr>
<td>3</td>
<td>0.1839</td>
<td>0.2507</td>
<td>0.0787</td>
<td>0.0557</td>
</tr>
</tbody>
</table>
NCL experiments (5)

• Hyper-parameter Tuning
NCL experiments (6)

• Applying NCL on Other GNN Backbones

<table>
<thead>
<tr>
<th>Method</th>
<th>MovieLens-1M</th>
<th>Yelp</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall@10</td>
<td>NDCG@10</td>
<td>Recall@10</td>
</tr>
<tr>
<td>NGCF</td>
<td>0.1846</td>
<td>0.2528</td>
<td>0.0630</td>
</tr>
<tr>
<td>+NCL</td>
<td>0.1852</td>
<td>0.2542</td>
<td>0.0663</td>
</tr>
<tr>
<td>DGCF</td>
<td>0.1853</td>
<td>0.2500</td>
<td>0.0723</td>
</tr>
<tr>
<td>+NCL</td>
<td>0.1877</td>
<td>0.2522</td>
<td>0.0739</td>
</tr>
<tr>
<td>LightGCN</td>
<td>0.1888</td>
<td>0.2526</td>
<td>0.0833</td>
</tr>
<tr>
<td>+NCL</td>
<td>0.2057</td>
<td>0.2732</td>
<td>0.0920</td>
</tr>
</tbody>
</table>
NCL experiments (7)

- Visualizing the Distribution of Representations

(a) LightGCN  (b) NCL  (c) LightGCN  (d) NCL

Figure 6: Visualization of item embeddings. Items from ML-1M and Yelp are illustrated in (a), (b) and (c), (d), respectively.
CORE:
Simple and Effective Session-based Recommendation within Consistent Representation Space

Yupeng Hou, Binbin Hu, Zhiqiang Zhang, Wayne Xin Zhao.
SIGIR 2022, short paper.
Background - Session-based Rec

- Next-item prediction;
- Anonymous sessions;
- Short-term Interest;
Background - Session-based Rec

• Existing studys - Fancy and complex session encoders

• RNN, CNN, GNN
• RNN + Attention, CNN + Attention, GNN + Attention
• Transformers
• GNN + Transformers

.... 😞
Observation

• Encoder-Decoder
Issue

• Inconsistent Prediction
• (a toy example)

sessions share a common objective
1. \langle a \rangle
2. \langle a, a \rangle
3. \langle a, a, a \rangle
4. \langle a, a, a, a \rangle
Idea

• What if encoding-decoding in consistent representation space mandatorily? 😐

• Basically, linear combination as encoder💡
Challenge

• Strong power of DNNs + consistent representation space;

• Prevent overfitting of item embeddings;
  (in consistent representation space)
COnsistent REpresentation - RCE

(Representation-Consistent Encoder)

\[ \alpha = \text{DNNs}([h_{s,1}; h_{s,2}; \ldots; h_{s,n}]) \]

\[ h_s = \sum_{i=1}^{n} \alpha_i h_{s,i}. \]

DNNs can be:
- Pooling;
- Transformers;
- … …
COnsistent REpresentation - RDM (Robust Distance Measuring)

• Traditional cross-entropy loss

\[ \ell_{\text{ori}} = - \log \frac{\exp(h_s \cdot h_{v^+})}{\sum_{i=1}^{m} \exp(h_s \cdot h_{v_i})} \]

\[ \propto \sum_{v^- \in \mathcal{V} \setminus \{v^+\}} \left( \|h_s - h_{v^+}\|^2 - \|h_s - h_{v^-}\|^2 + 2 \right). \]

\((N - 1)\) -tuplet loss with L2-distance & fixed margin 2
**COnsistent REpresentation - RDM**
(Robust Distance Measuring)

\((N - 1)\) – tuplet loss with L2-distance & fixed margin 2

\[
\ell = -\log \frac{\exp \left( \frac{\cos(h_s, h_v^+)}{\tau} \right)}{\sum_{i=1}^{m} \exp \left( \frac{\cos(h_s, h_{v_i}^-)}{\tau} \right)},
\]

(contrastive learning)
Sessions \(<->\) Next items

**Robust Distance Measuring**

\(h_s\) ← \(h_v^+\) ← \(h_v^-\)
CORE experiments

• 5 widely-used public datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Interactions</th>
<th># Items</th>
<th># Sessions</th>
<th>Avg. Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diginetica</td>
<td>786,582</td>
<td>42,862</td>
<td>204,532</td>
<td>4.12</td>
</tr>
<tr>
<td>Nowplaying</td>
<td>1,085,410</td>
<td>59,593</td>
<td>145,612</td>
<td>9.21</td>
</tr>
<tr>
<td>RetailRocket</td>
<td>871,637</td>
<td>51,428</td>
<td>321,032</td>
<td>6.40</td>
</tr>
<tr>
<td>Tmall</td>
<td>427,797</td>
<td>37,367</td>
<td>66,909</td>
<td>10.62</td>
</tr>
<tr>
<td>Yoochoose</td>
<td>1,434,349</td>
<td>19,690</td>
<td>470,477</td>
<td>4.64</td>
</tr>
</tbody>
</table>

• Carefully hyper-parameter tuning for all baselines

• https://anonymous.4open.science/r/CORE (temporarily)
## CORE experiments (1)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>FPMC</th>
<th>GRU4Rec</th>
<th>NARM</th>
<th>SR-GNN</th>
<th>NISER+</th>
<th>LESSR</th>
<th>SGNN-HN</th>
<th>SASRec</th>
<th>GC-SAN</th>
<th>CL4Rec</th>
<th>CORE-ave</th>
<th>CORE-trm</th>
<th>Improv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diginetica</td>
<td>R@20</td>
<td>31.83</td>
<td>45.43</td>
<td>47.68</td>
<td>48.76</td>
<td>51.23</td>
<td>48.80</td>
<td>50.89</td>
<td>49.86</td>
<td>50.95</td>
<td>50.03</td>
<td>50.21</td>
<td>52.89*</td>
<td>+3.24%</td>
</tr>
<tr>
<td></td>
<td>M@20</td>
<td>8.79</td>
<td>14.77</td>
<td>15.58</td>
<td>16.93</td>
<td>18.32</td>
<td>16.96</td>
<td>17.25</td>
<td>17.19</td>
<td>17.84</td>
<td>17.26</td>
<td>18.07</td>
<td>18.58*</td>
<td>+1.42%</td>
</tr>
<tr>
<td>Nowplaying</td>
<td>R@20</td>
<td>10.18</td>
<td>13.80</td>
<td>14.17</td>
<td>15.28</td>
<td>16.55</td>
<td>17.60</td>
<td>16.75</td>
<td>20.69</td>
<td>18.30</td>
<td>20.59</td>
<td>20.31</td>
<td>21.81*</td>
<td>+5.41%</td>
</tr>
<tr>
<td>RetailRocket</td>
<td>R@20</td>
<td>46.04</td>
<td>55.32</td>
<td>58.65</td>
<td>58.71</td>
<td>60.36</td>
<td>56.22</td>
<td>58.82</td>
<td>59.81</td>
<td>60.18</td>
<td>59.69</td>
<td>59.18</td>
<td>61.85*</td>
<td>+2.47%</td>
</tr>
<tr>
<td></td>
<td>M@20</td>
<td>21.95</td>
<td>33.18</td>
<td>34.69</td>
<td>36.42</td>
<td>37.43</td>
<td>37.11</td>
<td>35.72</td>
<td>36.03</td>
<td>36.85</td>
<td>35.95</td>
<td>37.52*</td>
<td>38.76*</td>
<td>+3.55%</td>
</tr>
<tr>
<td>Tmall</td>
<td>R@20</td>
<td>20.30</td>
<td>23.25</td>
<td>31.67</td>
<td>33.65</td>
<td>35.97</td>
<td>32.45</td>
<td>39.14</td>
<td>35.82</td>
<td>35.32</td>
<td>35.59</td>
<td>44.67*</td>
<td>44.48*</td>
<td>+14.13%</td>
</tr>
<tr>
<td></td>
<td>M@20</td>
<td>13.07</td>
<td>15.78</td>
<td>21.83</td>
<td>25.27</td>
<td>27.06</td>
<td>23.96</td>
<td>23.46</td>
<td>25.10</td>
<td>23.48</td>
<td>25.07</td>
<td>31.85*</td>
<td>31.72*</td>
<td>+17.70%</td>
</tr>
<tr>
<td>Yoochoose</td>
<td>R@20</td>
<td>–</td>
<td>60.78</td>
<td>61.67</td>
<td>61.84</td>
<td>62.99</td>
<td>62.89</td>
<td>62.49</td>
<td>63.55</td>
<td>63.24</td>
<td>63.61</td>
<td>58.83</td>
<td>64.61*</td>
<td>+1.57%</td>
</tr>
<tr>
<td></td>
<td>M@20</td>
<td>–</td>
<td>27.27</td>
<td>27.82</td>
<td>28.15</td>
<td>28.98</td>
<td>28.59</td>
<td>28.24</td>
<td>28.63</td>
<td>29.00</td>
<td>28.73</td>
<td>25.05</td>
<td>28.24</td>
<td>–</td>
</tr>
</tbody>
</table>
CORE experiments (2)

- Efficiency

Variant with only item embs
CORE experiments (3)

- Ablation Study

<table>
<thead>
<tr>
<th>Method</th>
<th>Diginetica</th>
<th></th>
<th>RetailRocket</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R@20</td>
<td>M@20</td>
<td>R@20</td>
<td>M@20</td>
</tr>
<tr>
<td>CORE</td>
<td>52.89</td>
<td>18.58</td>
<td>61.85</td>
<td>38.76</td>
</tr>
<tr>
<td>w/o RCE</td>
<td>49.82</td>
<td>17.41</td>
<td>59.59</td>
<td>36.27</td>
</tr>
<tr>
<td>w/o RDM</td>
<td>52.31</td>
<td>18.38</td>
<td>60.93</td>
<td>37.72</td>
</tr>
<tr>
<td>SASRec</td>
<td>49.86</td>
<td>17.19</td>
<td>59.81</td>
<td>36.03</td>
</tr>
</tbody>
</table>
CORE experiments (4)

- Improving existing methods with RCE & RDM

<table>
<thead>
<tr>
<th>Method</th>
<th>Diginetica</th>
<th>RetailRocket</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R@20</td>
<td>M@20</td>
</tr>
<tr>
<td>NARM</td>
<td>47.68</td>
<td>15.58</td>
</tr>
<tr>
<td>+ RCE</td>
<td>51.86</td>
<td>18.27</td>
</tr>
<tr>
<td>+ RDM</td>
<td>51.62</td>
<td>17.79</td>
</tr>
<tr>
<td>+ All</td>
<td>52.51</td>
<td>18.58</td>
</tr>
<tr>
<td>SR-GNN</td>
<td>48.76</td>
<td>16.93</td>
</tr>
<tr>
<td>+ RCE</td>
<td>49.51</td>
<td>17.53</td>
</tr>
<tr>
<td>+ RDM</td>
<td>51.36</td>
<td>18.57</td>
</tr>
<tr>
<td>+ All</td>
<td>52.38</td>
<td>18.95</td>
</tr>
</tbody>
</table>
CORE experiments (5)

- Visualization of session embeddings
- (sessions with same next-item are in the same class)

(a) GRU4Rec  (b) SASRec  (c) CORE (ours)
CORE experiments (6)

- Hyper-parameter Tuning

(a) Diginetica

(b) RetailRocket

(c) Diginetica

(d) RetailRocket
Conducting RecSys Research w/ RecBole
• [https://recbole.io/](https://recbole.io/)

• PyTorch, 78 models in 4 categories, 28 processed datasets

• One-stop solution for RecSys research 😊
config file

model with forward, calculate_loss, predict

trainer w/ EM
https://anonymous.4open.science/r/CORE

- **props**
- **README.md**
- **core_ave.py**
- **core_trm.py**
- **main.py**

- config file
- model with forward, calculate_loss, predict
GNN-enhanced RecSys?

- [https://github.com/RUCAIBox/RecBole-GNN](https://github.com/RUCAIBox/RecBole-GNN)
- 15 new models!
- Leaderboards for 3 categories;
- Efficient and reusable graph processing;

- Credit to Lanling and Changxin
Conclusion

Improving recommendation via Contrastive Learning

Yupeng Hou
Conclusion & QA

• Improving RecSys via CL
  • NCL [TheWebConf 22] for graph collaborative filtering;
  • CORE [SIGIR 22 short] for session-based recommendation;

• Self-supervised signals in RecSys
  • High-order neighbors (structural & semantic);
  • Sessions' next-item;

• Conducting Research with RecBole