Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding

Yushun Dong, Sihang Jiang, Simon Zhu, Hanzhi Zhou
ROADMAP

1. Background & Motivation
   - What problem to solve?
   - Why need a new model?

2. Methodology
   - What’s our model?
   - How does it solve the problem?

3. Experiment Design
   - How to evaluate?
   - What’re the datasets and baselines?

4. Results & Other Conclusions
   - How does our model perform?
   - Future works
Background & Motivation
Top-N sequential recommendation captures customers’ sequential patterns aside from general behavior

Customers’ General Preferences vs. Sequential Patterns

- e.g: Buying iPhone accessory after buying an IPhone.

What Top-N sequential recommendation does:

- recommends each user N items that maximize his/her future needs, by considering both general preferences and sequential patterns. Unlike conventional top-N recommendation, top-N sequential recommendation models the user behavior as a sequence of items, instead of a set of items.
Current sequential recommendation systems leave out union-level and skip patterns

- The current Markov Chain models only point-level sequential patterns.

Figure 1: An example of point and union level dynamic pattern influences, the order of Markov chain $L = 3$
Data exploration reveals common union-level and skip patterns

One Rule:

$$(S^u_{t-L}, \ldots, S^u_{t-2}, S^u_{t-1}) \rightarrow S^u_t.$$ 

(a) MovieLens  
(b) Gowalla

Figure 2: The number of association rules vs $L$ and skip steps. The minimum support count = 5 and the minimum confidence = 50%.
New model (Caser) gets the best of both worlds

ConvolutionAI Sequence Embedding Recommendation Model (Caser)

- Caser models both users’ general preferences and sequential patterns, and generalizes several existing state-of-the-art methods in a single unified framework.
- Caser capture sequential patterns at point-level, union-level, and of skip behaviors with horizontal and vertical convolutional filters.
Methodology
How can we build a better model?

Questions to be answered:

● How to model union-level influences?
● How to model point-level influences?
● How to incorporate skip behaviors in the model?
How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?

A basic intuition of union-level influence is that the **union exerts influence as a whole**, which is similar to the localized characteristic of an image.

![Diagram showing different arrangements of union levels](image-url)
How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?

Such similarity motivates us to utilize convolutional kernel (i.e., filters) to extract such localized union information.
How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?

Point-level can be modeled using traditional weighted-summing.

[Diagram showing sequence of variables $S_1^u, S_{t-3}^u, S_{t-2}^u, S_{t-1}^u, S_t^u, S_{t+1}^u, S_{|S^u|}^u$ and their embeddings and weights leading to prediction $S_t^u$.]
How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?

Point-level can be modeled using traditional weighted-summing.
How can we build a better model?

Process overview:

Embeddings

$S_{t-1}^u$  
$S_{t-2}^u$  
$S_{t-3}^u$  
$t$
How can we build a better model?

Process overview:
How can we build a better model?

Process overview:
How can we build a better model?

Process overview:

Embeddings

$S^u_{t-1}$

$S^u_{t-2}$

$S^u_{t-3}$

$t$
How can we build a better model?

Process overview:

Embeddings

$S_{t-3}^u$

$S_{t-2}^u$

$S_{t-1}^u$

$t$

$S_t^u$
How can we build a better model?

Process overview:

![Diagram showing a process overview related to building a better model.](image)
How can we build a better model?

Process overview:
How can we build a better model?

Process overview:
How can we build a better model?

Process overview:

Embeddings

$S^u_{t-3}$  
$S^u_{t-2}$  
$S^u_{t-1}$  
$t$
How can we build a better model?

Process overview:

Embeddings

$S_{t-3}^u$

$S_{t-2}^u$

$S_{t-1}^u$

$t$
How can we build a better model?

Process overview:
How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?

The solution can be straightforward: directly turn predicting the next item into predicting the next item set (including T items).

\[
p(S|\Theta) = \prod_u \prod_t \sigma(y_{S_t}^{(u,t)}) \prod_{j \neq S_t} (1 - \sigma(y_j^{(u,t)}))
\]

\[
\ell = \sum_u \sum_{t \in C^u} \sum_{i \in D_t^u} -\log(\sigma(y_i^{(u,t)})) + \sum_{j \neq i} -\log(1 - \sigma(y_j^{(u,t)}))
\]

\[
D_t^u = \{S_t^u, S_{t+1}^u, ..., S_{t+T}^u\}
\]
\[
C^u = \{L+1, L+2, ..., |S^u|\}
\]
Experiment Design
Datasets & Train/Test Split

Datasets:

**MovieLens**: user id, movie, rating

**Gowalla**: user id, check-in time, location

**Foursquare (non-public)**: user id, check-in time, location

**Tmall (non-public)**: user id, purchase time, item

Selected because of acceptable “sequential intensity” according to the authors

Train/test Split:

70% train, 10% validation (tune), 20% test
Dataset Characteristics

Table 1: Statistics of the datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Sequential Intensity</th>
<th>#users</th>
<th>#items</th>
<th>avg. actions per user</th>
<th>Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLens</td>
<td>0.3265</td>
<td>6.0k</td>
<td>3.4k</td>
<td>165.50</td>
<td>95.16%</td>
</tr>
<tr>
<td>Gowalla</td>
<td>0.0748</td>
<td>13.1k</td>
<td>14.0k</td>
<td>40.74</td>
<td>99.71%</td>
</tr>
<tr>
<td>Foursquare</td>
<td>0.0378</td>
<td>10.1k</td>
<td>23.4k</td>
<td>30.16</td>
<td>99.87%</td>
</tr>
<tr>
<td>Tmall</td>
<td>0.0104</td>
<td>23.8k</td>
<td>12.2k</td>
<td>13.93</td>
<td>99.89%</td>
</tr>
</tbody>
</table>

Define a rule $X_L \rightarrow Y$ to be $(S_{t-L}, ..., S_t) \rightarrow S_{t+1}$

$X_L : L$ previous items, $Y :$ next item

$$#rules = \sum_{L=1}^{5} \left| \left\{ X_L \mid \frac{sup(X_L Y)}{sup(X_L)} > 0.5 \text{ and } sup(X_L Y) \geq 5 \right\} \right|$$

where $sup(X_L Y)$(support) is the number of rules that have $Y$ follows $X_L$ and $sup(X_L)$ is the number of occurrences of $X_L$

Sequential Intensity $(SI) = \frac{#rules}{#users}$

A measure of the intensity of sequential patterns
Evaluation Metrics

Rhat: Predicted top N items the user want next

R: Ground truth (actual items the user want next)

Rel(N): 0 or 1, indicates whether the item predicted N exists in the ground truth

\[
\text{Prec@N} = \frac{|R \cap \hat{R}_{1:N}|}{N},
\]

\[
\text{Recall@N} = \frac{|R \cap \hat{R}_{1:N}|}{|R|}.
\]

\[
\text{AP} = \frac{\sum_{N=1}^{\hat{R}} |\hat{R}| \text{Prec@N} \times \text{rel}(N)}{|\hat{R}|},
\]
Results & Other Conclusions
Caser performs better in most cases with different metrics.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>POP</th>
<th>BPR</th>
<th>FMC</th>
<th>PPMC</th>
<th>Fossil</th>
<th>GRU4Rec</th>
<th>Caser</th>
<th>Improv.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec@1</td>
<td>0.1280</td>
<td>0.1478</td>
<td>0.1748</td>
<td>0.2022</td>
<td>0.2306</td>
<td><strong>0.2515</strong></td>
<td>0.2502</td>
<td>-0.5%</td>
</tr>
<tr>
<td></td>
<td>Prec@5</td>
<td>0.1113</td>
<td>0.1288</td>
<td>0.1505</td>
<td>0.1659</td>
<td>0.2000</td>
<td>0.2146</td>
<td><strong>0.2175</strong></td>
<td>1.4%</td>
</tr>
<tr>
<td></td>
<td>Prec@10</td>
<td>0.1011</td>
<td>0.1193</td>
<td>0.1317</td>
<td>0.1460</td>
<td>0.1806</td>
<td>0.1916</td>
<td><strong>0.1991</strong></td>
<td>4.0%</td>
</tr>
<tr>
<td></td>
<td>Recall@1</td>
<td>0.0050</td>
<td>0.0070</td>
<td>0.0104</td>
<td>0.0118</td>
<td>0.0144</td>
<td><strong>0.0153</strong></td>
<td>0.0148</td>
<td>-3.3%</td>
</tr>
<tr>
<td></td>
<td>Recall@5</td>
<td>0.0213</td>
<td>0.0312</td>
<td>0.0432</td>
<td>0.0468</td>
<td>0.0602</td>
<td>0.0629</td>
<td><strong>0.0632</strong></td>
<td>0.5%</td>
</tr>
<tr>
<td></td>
<td>Recall@10</td>
<td>0.0375</td>
<td>0.0560</td>
<td>0.0722</td>
<td>0.0777</td>
<td>0.1061</td>
<td>0.1093</td>
<td><strong>0.1121</strong></td>
<td>2.6%</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>0.0687</td>
<td>0.0913</td>
<td>0.0949</td>
<td>0.1053</td>
<td>0.1354</td>
<td>0.1440</td>
<td><strong>0.1507</strong></td>
<td>4.7%</td>
</tr>
<tr>
<td>MovieLens</td>
<td>Prec@1</td>
<td>0.0517</td>
<td>0.1640</td>
<td>0.1532</td>
<td>0.1555</td>
<td>0.1736</td>
<td>0.1736</td>
<td><strong>0.1961</strong></td>
<td>13.0%</td>
</tr>
<tr>
<td></td>
<td>Prec@5</td>
<td>0.0362</td>
<td>0.0983</td>
<td>0.0876</td>
<td>0.0936</td>
<td>0.1045</td>
<td>0.1072</td>
<td><strong>0.1129</strong></td>
<td>8.0%</td>
</tr>
<tr>
<td></td>
<td>Prec@10</td>
<td>0.0281</td>
<td>0.0726</td>
<td>0.0657</td>
<td>0.0698</td>
<td>0.0782</td>
<td>0.0751</td>
<td><strong>0.0833</strong></td>
<td>6.5%</td>
</tr>
<tr>
<td></td>
<td>Recall@1</td>
<td>0.0064</td>
<td>0.0250</td>
<td>0.0234</td>
<td>0.0256</td>
<td>0.0277</td>
<td>0.0155</td>
<td><strong>0.0310</strong></td>
<td>11.9%</td>
</tr>
<tr>
<td></td>
<td>Recall@5</td>
<td>0.0257</td>
<td>0.0743</td>
<td>0.0648</td>
<td>0.0722</td>
<td>0.0793</td>
<td>0.0529</td>
<td><strong>0.0845</strong></td>
<td>6.6%</td>
</tr>
<tr>
<td></td>
<td>Recall@10</td>
<td>0.0402</td>
<td>0.1077</td>
<td>0.0950</td>
<td>0.1059</td>
<td>0.1166</td>
<td>0.0826</td>
<td><strong>0.1223</strong></td>
<td>4.9%</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>0.0229</td>
<td>0.0767</td>
<td>0.0711</td>
<td>0.0764</td>
<td>0.0848</td>
<td>0.0580</td>
<td><strong>0.0928</strong></td>
<td>9.4%</td>
</tr>
<tr>
<td>Gowalla</td>
<td>Prec@1</td>
<td>0.1090</td>
<td>0.1233</td>
<td>0.0875</td>
<td>0.1081</td>
<td>0.1191</td>
<td>0.1018</td>
<td><strong>0.1351</strong></td>
<td>13.4%</td>
</tr>
<tr>
<td></td>
<td>Prec@5</td>
<td>0.0477</td>
<td>0.0543</td>
<td>0.0445</td>
<td>0.0555</td>
<td>0.0580</td>
<td>0.0475</td>
<td><strong>0.0619</strong></td>
<td>6.7%</td>
</tr>
<tr>
<td></td>
<td>Prec@10</td>
<td>0.0304</td>
<td>0.0348</td>
<td>0.0309</td>
<td>0.0385</td>
<td>0.0399</td>
<td>0.0331</td>
<td><strong>0.0425</strong></td>
<td>6.5%</td>
</tr>
<tr>
<td></td>
<td>Recall@1</td>
<td>0.0376</td>
<td>0.0445</td>
<td>0.0305</td>
<td>0.0440</td>
<td>0.0497</td>
<td>0.0369</td>
<td><strong>0.0565</strong></td>
<td>13.7%</td>
</tr>
<tr>
<td></td>
<td>Recall@5</td>
<td>0.0800</td>
<td>0.0888</td>
<td>0.0689</td>
<td>0.0959</td>
<td>0.0948</td>
<td>0.0770</td>
<td><strong>0.1035</strong></td>
<td>7.9%</td>
</tr>
<tr>
<td></td>
<td>Recall@10</td>
<td>0.0954</td>
<td>0.1061</td>
<td>0.0911</td>
<td>0.1200</td>
<td>0.1187</td>
<td>0.1011</td>
<td><strong>0.1291</strong></td>
<td>7.6%</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>0.0636</td>
<td>0.0719</td>
<td>0.0571</td>
<td>0.0782</td>
<td>0.0823</td>
<td>0.0643</td>
<td><strong>0.0909</strong></td>
<td>10.4%</td>
</tr>
<tr>
<td>Foursquare</td>
<td>Prec@1</td>
<td>0.0010</td>
<td>0.0111</td>
<td>0.0197</td>
<td>0.0210</td>
<td>0.0280</td>
<td>0.0139</td>
<td><strong>0.0312</strong></td>
<td>11.4%</td>
</tr>
<tr>
<td></td>
<td>Prec@5</td>
<td>0.0009</td>
<td>0.0081</td>
<td>0.0114</td>
<td>0.0120</td>
<td>0.0149</td>
<td>0.0090</td>
<td><strong>0.0179</strong></td>
<td>20.1%</td>
</tr>
<tr>
<td></td>
<td>Prec@10</td>
<td>0.0007</td>
<td>0.0063</td>
<td>0.0084</td>
<td>0.0090</td>
<td>0.0104</td>
<td>0.0070</td>
<td><strong>0.0132</strong></td>
<td>26.9%</td>
</tr>
<tr>
<td></td>
<td>Recall@1</td>
<td>0.0004</td>
<td>0.0046</td>
<td>0.0079</td>
<td>0.0082</td>
<td>0.0117</td>
<td>0.0056</td>
<td><strong>0.0130</strong></td>
<td>11.1%</td>
</tr>
<tr>
<td></td>
<td>Recall@5</td>
<td>0.0019</td>
<td>0.0169</td>
<td>0.0226</td>
<td>0.0245</td>
<td>0.0306</td>
<td>0.0180</td>
<td><strong>0.0366</strong></td>
<td>19.6%</td>
</tr>
<tr>
<td></td>
<td>Recall@10</td>
<td>0.0026</td>
<td>0.0260</td>
<td>0.0333</td>
<td>0.0364</td>
<td>0.0425</td>
<td>0.0278</td>
<td><strong>0.0534</strong></td>
<td>25.6%</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>0.0030</td>
<td>0.0145</td>
<td>0.0197</td>
<td>0.0212</td>
<td>0.0256</td>
<td>0.0164</td>
<td><strong>0.0310</strong></td>
<td>21.1%</td>
</tr>
</tbody>
</table>
Caser components

Table 3: MAP vs. Caser Components

<table>
<thead>
<tr>
<th></th>
<th>MovieLens</th>
<th>Gowalla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caser-p</td>
<td>0.0935</td>
<td>0.0777</td>
</tr>
<tr>
<td>Caser-h</td>
<td>0.1304</td>
<td>0.0805</td>
</tr>
<tr>
<td>Caser-v</td>
<td>0.1403</td>
<td>0.0841</td>
</tr>
<tr>
<td>Caser-vh</td>
<td>0.1448</td>
<td>0.0856</td>
</tr>
<tr>
<td>Caser-ph</td>
<td>0.1372</td>
<td>0.0911</td>
</tr>
<tr>
<td>Caser-pv</td>
<td>0.1494</td>
<td>0.0921</td>
</tr>
<tr>
<td>Caser-pvh</td>
<td>0.1507</td>
<td>0.0928</td>
</tr>
</tbody>
</table>

Different components of Caser model (such as horizontal and vertical filters) all exhibit effectiveness.
Numerical Results

Recent items have higher weight.
Capture of union-level sequential patterns

(a) Previous Sequence

\[ S_1 \quad S_2 \quad S_3 \quad S_4 \quad S_5 \]

(b) Predictions

\[ \hat{R}_1 \quad \hat{R}_2 \quad \hat{R}_3 \]

<table>
<thead>
<tr>
<th>Masking Items</th>
<th>New Rank of $\hat{R}_3$ after masking</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1, S_2$</td>
<td>2</td>
</tr>
<tr>
<td>$S_3$</td>
<td>32</td>
</tr>
<tr>
<td>$S_4$</td>
<td>117</td>
</tr>
<tr>
<td>$S_5$</td>
<td>77</td>
</tr>
<tr>
<td>$S_3, S_4, S_5$</td>
<td>513</td>
</tr>
</tbody>
</table>

Figure 8: Horizontal convolutional filters’s effectiveness of capturing union-level sequential patterns on MovieLens data.
Thank you for listening!