Recommendation with Sequences of Micro Behaviors

Mingkai He (revised by Weike Pan)

College of Computer Science and Software Engineering
Shenzhen University

Sequential Heterogeneous One-Class Collaborative Filtering (SHOCCF)

- **Input:** A historical heterogeneous sequence
  \[ S_u = \{(i_{u}^{t-L+1}, f_{u}^{t-L+1}), \ldots, (i_{u}^{\ell}, f_{u}^{\ell}), \ldots, (i_{u}^{t}, f_{u}^{t})\}, \]
  where \((i_{u}^{t}, f_{u}^{t})\) denotes the (item, behavior) pair at timestamp \(t\) w.r.t. user \(u\).

- **Goal:** Predict the next likely-to-purchase item \(i\) of a user \(u\) from \(\mathcal{I}\) at timestamp \(t + 1\).
Main Idea

- Recommendation with sequences of micro behaviors (RIB)
  - An input layer that concatenates item embedding and behavior embedding
  - A GRU layer that obtains the sequential information
  - An attention layer that simulates the effects of different micro-behaviors in the sequence on recommendation
**Table:** Some notations and explanations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{U}$</td>
<td>the whole set of users</td>
</tr>
<tr>
<td>$\mathcal{I}$</td>
<td>the whole set of items</td>
</tr>
<tr>
<td>$\mathcal{F}$</td>
<td>the whole set of feedback (or behaviors)</td>
</tr>
<tr>
<td>$u \in \mathcal{U}$</td>
<td>user ID</td>
</tr>
<tr>
<td>$i \in \mathcal{I}$</td>
<td>item ID</td>
</tr>
<tr>
<td>$f \in \mathcal{F}$</td>
<td>behavior ID</td>
</tr>
<tr>
<td>$\mathcal{S}_u$</td>
<td>the sequence of (item, behavior) pairs that user $u$ has interacted with</td>
</tr>
<tr>
<td>$i^t_u \in \mathcal{I}$</td>
<td>the item interacted by user $u$ at timestamp $t$</td>
</tr>
<tr>
<td>$f^t_u \in \mathcal{F}$</td>
<td>the behavior of user $u$ at timestamp $t$</td>
</tr>
</tbody>
</table>
### Table: Some notations and explanations (cont.).

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_i \in \mathbb{R}^{d \times 1}$</td>
<td>the embedding of item $i$</td>
</tr>
<tr>
<td>$F_f \in \mathbb{R}^{d \times 1}$</td>
<td>the embedding of behavior $f$</td>
</tr>
<tr>
<td>$\hat{r}_{t+1,j}$</td>
<td>the preference score on item $j$ at timestamp $t + 1$</td>
</tr>
</tbody>
</table>
Recommendation with sequences of micro behaviors (RIB)

**Figure:** The network architecture of RIB. Firstly, we obtain a feature vector with item embedding and behavior embedding by the input layer. Secondly, we use a GRU layer to capture the sequential information. Thirdly, we use an attention layer to simulate the effects of different micro-behaviors on recommendation. Finally, we use an output layer to aggregate the feature of each timestamp to obtain a representation of the heterogeneous sequence.
For each item $i_u^t$ and the corresponding behavior $f_u^t$ of user $u$ in a heterogeneous sequence $S_u$, we use a concatenate operation to obtain the feature $e_t$:

$$e_t = concatenate(V_{i_u^t}, F_{f_u^t})$$  \hspace{1cm} (1)

where $V_{i_u^t} \in \mathbb{R}^{d \times 1}$ and $F_{f_u^t} \in \mathbb{R}^{d \times 1}$ denotes the embedding of item $i_u^t$ and behavior $f_u^t$, respectively.

RIB distinguishes different types of behaviors by adding the behavior information to the input of GRU.
At timestamp $t$, GRU [Hidasi and Karatzoglou, 2018] takes $e_t$ and the hidden state $h_{t-1}$ of timestamp $t-1$ as input, and output the current hidden state $h_t$:

$$r_t = \sigma(W_{er}e_t + W_{hr}h_{t-1})$$  \hspace{1cm} (2)
$$z_t = \sigma(W_{ez}e_t + W_{hz}h_{t-1})$$  \hspace{1cm} (3)
$$c_t = \tanh(W_{ec}e_t + W_{hc}(r_th_{t-1}))$$  \hspace{1cm} (4)
$$h_t = (1 - z_t)h_{t-1} + z_tc_t$$  \hspace{1cm} (5)

where $W_{er}, W_{ez}, W_{ec} \in \mathbb{R}^{d \times 2d}$ and $W_{hr}, W_{hz}, W_{hc} \in \mathbb{R}^{d \times d}$ are the trainable parameters of GRU. $\sigma$ and $\tanh$ denote the sigmoid activation function and the tanh activation function, respectively. $r_t \in \mathbb{R}^{d \times 1}$ is the output of the reset gate, $z_t \in \mathbb{R}^{d \times 1}$ is the output of the update gate, $c_t \in \mathbb{R}^{d \times 1}$ is the output of the memory cell, and $h_t \in \mathbb{R}^{d \times 1}$ is the hidden state of timestamp $t$. 


Attention Layer

In order to assign appropriate weight to the hidden state at each timestamp, we use an attention layer to simulate the different effects of different micro-behaviors in the sequence on recommendation:

\[ M_t = \tanh(W_m h_t + b_m) \]  \hspace{1cm} (6)

\[ \alpha_t = \text{softmax}(W_\alpha M_t + b_\alpha) \]  \hspace{1cm} (7)

where \( W_m \in \mathbb{R}^{d \times d} \), \( b_m \in \mathbb{R}^{d \times 1} \), \( M_t \in \mathbb{R}^{d \times 1} \), \( W_\alpha \in \mathbb{R}^{d \times d} \) and \( b_\alpha \in \mathbb{R}^{d \times 1} \) are the trainable parameters of the attention layer.

There are two aspects of the importance of the attention layer. Firstly, it simulates the different effects of different micro-behaviors in the sequence on recommendation. Secondly, the attention-based algorithm has better interpretability.
Output Layer

- We use a weighted summation to aggregate the feature of each timestamp, and then get the output vector $h_{RIB}$:

$$h_{RIB} = \frac{1}{|S_u|} \sum_{t=1}^{|S_u|} \alpha_t h_t$$  \hspace{1cm} (8)

- Finally, we can use a fully connected layer to get the prediction score for each item:

$$\hat{r} = \text{softmax}(W_{RIB}h_{RIB} + b_{RIB})$$  \hspace{1cm} (9)

where $W_{RIB} \in \mathbb{R}^{|I| \times d}$ and $b_{RIB} \in \mathbb{R}^{|I| \times 1}$ are trainable parameters.
Loss Function

The objective function of RIB with the commonly used cross-entropy loss function is as follows:

$$\mathcal{L} = - \sum_{s \in S} \sum_{j \in I} y_{sj} \log(\hat{r}_{t+1,j}) + (1 - y_{sj}) \log(1 - \hat{r}_{t+1,j})$$  \hspace{1cm} (10)

where $y_{sj} = 1$ only if item $j$ is the real interacted item of the sequence $s$ at timestamp $t + 1$, and $y_{sj} = 0$ otherwise.
Methods

Dataset

We preprocess the ML1M dataset as follows: 1) we keep the (user, item) pairs with a rating value equal to 5 as the purchase behaviors, and the rest as the examination behaviors; 2) we discard later duplicated (user, item, behavior) triples in a sequence; 3) we discard unpopular items that are purchased fewer than 5 times; 4) we remove a sequence in which the number of purchases is smaller than 5; and 5) for each interaction sequence, we take the last two purchases as the validation data and the test data, and the remaining as the training data.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Users</th>
<th># Items</th>
<th># Examinations</th>
<th># Purchases</th>
<th>Avg. Length</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML1M</td>
<td>5,645</td>
<td>2,357</td>
<td>628,892</td>
<td>223,305</td>
<td>150.96</td>
<td>6.41%</td>
</tr>
</tbody>
</table>

Table: Statistics of the processed dataset used in the experiments.
We evaluate the top-k recommendation performance via two commonly used ranking-oriented metrics, i.e., hit ratio (HR@10) and normalized discounted cumulative gain (NDCG@10). HR@k represents the hit ratio of a user’s target item in the recommended top-k items, which is used to measure the accuracy of the recommendation algorithm, while NDCG@k is more concerned about the ranking position of the user’s preferred target item in the top-k recommendation list.
For RIB, we set the embedding size as 64, the sequence length $L$ as 50, the batch size as 128, the learning rate as 0.001 and the dropout rate as 0.2.
Extension of RIB: RIB++

Figure: The network architecture of RIB++ [Zhuoxin Zhan and Ming, ]. We use $\text{trans}(f^t_u, f^{t+1}_u) \in \mathbb{R}^{d \times 1}$ to represent the transition between the current behavior $f^t_u$ and the next behavior $f^{t+1}_u$, and then add it to the $F^t_{f_u}$. There are four types of transitions in our experiments, i.e., e2e, e2p, p2e and p2p.
Table: Recommendation performance of RIB and RIB++ on ML1M.

<table>
<thead>
<tr>
<th>Method</th>
<th>ML1M</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HR@10</td>
</tr>
<tr>
<td>RIB</td>
<td>0.1302</td>
</tr>
<tr>
<td>RIB++</td>
<td>0.1461</td>
</tr>
</tbody>
</table>
By adding the behavior embedding to the input of the RNN-based method, it can be applied to SHOCCF with different types of behaviors.
Recurrent neural networks with top-k gains for session-based recommendations.

Micro behaviors: A new perspective in e-commerce recommender systems.

Zhuoxin Zhan, Mingkai He, W. P. and Ming, Z.
Transrec++: Translation-based sequential recommendation with heterogeneous feedback.
In *Frontiers of Computer Science, accepted on October 18, 2021.*