Incremental Learning for Multi-Interest Sequential Recommendation
Zhikai Wang (Cloudcatcher.888@sjtu.edu.cn)
Yanyan Shen (shenyy@sjtu.edu.cn)
Shanghai Jiao Tong University
Outline

- Background
- Problem Setting
- The IMSR Framework
- Experiments
- Conclusion
Background
What is Multi-interest Sequential Recommendation?

- **Sequential recommendation (SR):**
  - Typically extract one user preference vector from the interaction sequence.

- **Multi-interest sequential recommendation (MSR):**
  - Recent works argue that user usually has multiple interests beneath interaction sequence.

  **Explicitly** compute multiple interest vectors based on user interaction sequence.

---

**MSR model** is testified to be good at handling the billion-scale of items in platforms e.g., Taobao, Amazon. [1,2]

Do MSR Models Need to be Updated Incrementally? YES

- User interaction sequence \( S_u^1, S_u^2, \ldots, S_u^t, \ldots \) is collected continuously.
- Existing interests may drift & new interests may be generated.
- Retraining MSR model \( M_t \) on all historical \( S_u^1, S_u^2, \ldots, S_u^{t-1} \) & new data \( S_u^t \) is time-consuming.

\[ \text{Update an MSR model using new collected data } S_u^t. \]

Multi-interest vectors

<table>
<thead>
<tr>
<th>Time span ( t-2 )</th>
<th>Time span ( t-1 )</th>
<th>Time span ( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR Model ( M_{t-2} )</td>
<td>MSR Model ( M_{t-1} )</td>
<td>MSR Model ( M_t )</td>
</tr>
<tr>
<td>Item sequence</td>
<td>Item sequence</td>
<td>Item sequence</td>
</tr>
<tr>
<td>( S_u^{t-2} )</td>
<td>( S_u^{t-1} )</td>
<td>( S_u^t )</td>
</tr>
</tbody>
</table>

Background ➔ Problem Setting ➔ The IMSR Framework ➔ Experiments
How to Update an MSR Model Incrementally?

- **Design goals:**
  - **Retain** and slightly drift existing interest vectors to match with the newly collected interactions.
  - **Detect** the occurrence of new interests in each time span.
  - **Expand** new interests adaptively in each time span.
### Why Existing Incremental Learning Methods Fail?

<table>
<thead>
<tr>
<th>Type</th>
<th>Main idea</th>
<th>Limitations</th>
<th>Retaining</th>
<th>Detection</th>
<th>Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reservoir-based methods (e.g., [1,2])</td>
<td>Select historical samples from the reservoir for model updating based on prioritizing recency or the extent of being forgotten</td>
<td>High cost</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Model-based methods (e.g., [3,4])</td>
<td>Enforce regularization terms, which regularly restrains the model parameters rather than user latent representations</td>
<td>Fixed model capacity</td>
<td>√</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Model-expansion methods (e.g., [5,6])</td>
<td>Expand the model capacity to cope with new knowledge during incremental learning</td>
<td>Low self-adaptivity</td>
<td>√</td>
<td>√</td>
<td>×</td>
</tr>
</tbody>
</table>

Problem Setting
Problem Setting

- **Notations** (in time span $t$):

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_u^t = {i_{1,u}^t, i_{2,u}^t, \ldots, i_{n,u}^t}$</td>
<td>User $u$’s historical interaction sequence</td>
</tr>
<tr>
<td>$i_{a,u}^t$ and $e_{a,u}^t$</td>
<td>Target item and its embedding</td>
</tr>
<tr>
<td>$H_u^t = {h_{1,u}^t, h_{2,u}^t, \ldots, h_{K,u}^t}$</td>
<td>Multi-interest user representation with $K$ interests</td>
</tr>
</tbody>
</table>

- **Incremental MSR**:
  - In each time span $t$, we aim to learn an MSR model $M_t$ which can map a user’s interaction sequence $S_u^t$ into a multi-interest user representation $H_u^t$:
    \[
    M_t, H_u^t \leftarrow \text{Incremental MSR}(S_u^t, M_{t-1}, H_u^{t-1})
    \]
The IMSR Framework
The IMSR Framework

Training procedure:
Phase 1: Detect interests with NID.
Phase 2: Retain existing interests with EIR.
Phase 3: Update model on a joint loss.
Phase 4: Trim redundant interests with PIT.
Phase 5: Concatenate interests.

Three modules for incremental learning:
- existing-interests retainer (EIR)
- new-interests detector (NID)
- projection-based interests trimmer (PIT)

MSR model $M_t$:
a dynamic routing or self-attention-based MSR model.
Existing-interests Retainer (EIR)

- Encourage the inner-product between each existing interest $h_k^t$ at time span $t$ and target item embedding $e_a^t$ to be close to that between existing interest $h_k^{t-1}$ at time span $t-1$ and $e_a^t$ using cross entropy loss [1]:

$$L_{k,u}^t = L_{CE}(\sigma\left(\frac{e_a^th_k^T}{\tau}\right), \sigma\left(\frac{e_a^th_k^{t-1}^T}{\tau}\right))$$

- Sum over all existing interests $H_k^t$ to obtain the total EIR loss:

$$L_{EIR}^t = \sum_{s_t \in S_t} \sum_{k=1}^{|H_k^t|} L_{k,u}^t$$

New-interests Detector (NID)

- Calculate the probability vector \([s_{i,1}, \ldots, s_{i,K^t}]\) of item \(i\) belonging to all existing interests \(h_{1}^{t-1}, \ldots, h_{K^t}^{t-1}\) using inner product \(s_{i,k} = e_i h_k^{t-1\top}\)

- Intuitively, the uncertainty of item \(i\) belonging to existing interests is proportional to the uniformity of \([s_{i,1}, \ldots, s_{i,K^t}]\) and can be measured by Query Sparsity Measurement[1]:

\[
\text{Uncertainty}(i) = \frac{1}{K^t} \sum_{k=1}^{K^t} s_{i,k} - \ln \sum_{k=1}^{K^t} e^{s_{i,k}} + \ln K^t
\]

- New interests are detected if the average uncertainty of items in \(S_u^t\) exceeds a threshold \(c_1\):

\[
\text{Uncertainty}(i) > c_1, i \in S_u^t
\]

Projection-based Interest Trimmer (PIT)

- **Step 1: Obtain new interests via projection:**
  - Learn $\delta K$ new interests for each user (omit $u, t$):
    \[ H^{new} = (h_1^{new}, ..., h_{\delta K}^{new}) \]
  - Project new interests on existing interests hyperplane:
    \[ h_k^{proj} = M_{exist} M_{exist} (M_{exist} M_{exist}^T)^{-1} h_k^{new} \]
  - Preserve the **orthogonal** part:
    \[ \Delta h_k^{new} = h_k^{new} - h_k^{proj} \]

- **Step 2: Trimming insignificant interests:**
  - Remove new interest vectors with **small** L2-norm:
    \[ \|\Delta h_k^{new}\|_{L2} < c_2 \]
Apply IMSR to Incrementally Update Two MSR Models

- **Dynamic routing based MSR (DR):**

- **Self-attention based MSR (SA):**
Training and Inference Procedure

- **Training in time span** $t$:
  - update the model parameters using $L^t_{SS}$ and $L^t_{EIR}$:
    \[
    L^t_{total} = L^t_{SS} + L^t_{EIR}
    \]
  - **Label aware attention**
    \[
    v^t_u = \sum_{k=1}^{K^t_u} \text{softmax} \left( e^t_a h^t_k \right) h^t_k
    \]
  - **Sample-softmax**
    \[
    L^t_{SS} = \sum_{s^t_u \in S^t} \log \frac{\exp(v^t_u e^t_a)}{\sum_{i \in I'} \exp(v^t_u e^t_i)}
    \]
    ($I' \subset I \setminus \{i^t_a\}$ is a small sampled subset of $I$)

- **Multi-interest vectors**
  - $h^t_1$
  - $h^t_2$
  - $\vdots$
  - $h^t_{K^t}$

- **Label-aware attention**

- **Sample-softmax**
  - $v^t_u$

- **Output**
  - $L^t_{SS}$
  - $L^t_{EIR}$
  - $L^t_{total}$
Training and Inference Procedure

- **Inference in time span t:**
  - Allocate the top-N items according to inner-product score $h^t_1 e^t_\alpha$ for each interest.
  - Union the item lists allocated by each interests to obtain the final recommendation list.

```
Multi-interest vectors

$h_1^t$  Item list 1  …

$h_2^t$  Item list 2  …

…

$h_K^t$  Item list $K$  …

Final recommendation list
```

```
...  …  …  …
```
Experiments
Experiment Settings

- **Datasets & Splitting:**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#users</th>
<th>#items</th>
<th>pre-training</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics</td>
<td>87,912</td>
<td>234,621</td>
<td>1,689,188</td>
<td>224,421</td>
<td>428,149</td>
<td>329,194</td>
<td>129,482</td>
<td>481,491</td>
<td>196,451</td>
</tr>
<tr>
<td>Clothing</td>
<td>285,464</td>
<td>376,859</td>
<td>5,748,920</td>
<td>864,371</td>
<td>574,922</td>
<td>957,329</td>
<td>1,134,792</td>
<td>943,422</td>
<td>1,274,084</td>
</tr>
<tr>
<td>Books</td>
<td>459,133</td>
<td>313,966</td>
<td>8,898,041</td>
<td>1,345,234</td>
<td>1,324,545</td>
<td>1,852,324</td>
<td>1,593,281</td>
<td>1,349,281</td>
<td>1,433,376</td>
</tr>
<tr>
<td>Taobao</td>
<td>976,779</td>
<td>1,708,530</td>
<td>85,384,110</td>
<td>12,329,481</td>
<td>14,481,123</td>
<td>22,129,123</td>
<td>9,329,128</td>
<td>14,238,129</td>
<td>12,877,126</td>
</tr>
</tbody>
</table>

- **Evaluation metrics:** Hit Ratio, NDCG on Top 20

- **Base models:** DR models: MIND [1], ComiRec-DR [2], SA model: ComiRec-SA [2].

- **Compared Learning Methods:** Full Retraining (FR), Fine-tuning (FT), SML (model-based IL, 2021), ADER (reservoir-based IL, 2020).

- **Hyperparameter setting:** $K_u^0 = 4$, $\delta K = 3$, $c_1 = 0.6 \text{(Taobao)}/0.4 \text{(Else)}$, $c_2 = 0.3$

Performance Comparison

<table>
<thead>
<tr>
<th>Base model</th>
<th>Learning method</th>
<th>Electronics</th>
<th></th>
<th></th>
<th>Clothing</th>
<th></th>
<th></th>
<th>Books</th>
<th></th>
<th></th>
<th>Taobao</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MIND</td>
<td>FR</td>
<td>16.03</td>
<td>16.43</td>
<td>11.15</td>
<td>16.23</td>
<td>15.98</td>
<td>10.57</td>
<td>13.82</td>
<td>11.95</td>
<td>10.47</td>
<td>43.29</td>
<td>24.90</td>
</tr>
<tr>
<td></td>
<td>FT</td>
<td>14.75</td>
<td>14.46</td>
<td>-</td>
<td>14.45</td>
<td>14.68</td>
<td>-</td>
<td>12.34</td>
<td>10.98</td>
<td>-</td>
<td>42.09</td>
<td>24.35</td>
</tr>
<tr>
<td></td>
<td>SML</td>
<td>15.41</td>
<td>15.17</td>
<td>4.71</td>
<td>15.27</td>
<td>14.81</td>
<td>3.26</td>
<td>13.12</td>
<td>11.12</td>
<td>3.97</td>
<td>42.88</td>
<td>24.58</td>
</tr>
<tr>
<td></td>
<td>ADER</td>
<td>15.64</td>
<td>14.98</td>
<td>4.84</td>
<td>15.62</td>
<td>15.20</td>
<td>5.76</td>
<td>12.97</td>
<td>11.48</td>
<td>4.64</td>
<td>42.90</td>
<td>24.24</td>
</tr>
<tr>
<td></td>
<td>IMSR</td>
<td>15.81*</td>
<td>15.71*</td>
<td>7.93</td>
<td>15.81*</td>
<td>15.71*</td>
<td>8.19</td>
<td>13.99*</td>
<td>11.94*</td>
<td>11.18</td>
<td>43.94*</td>
<td>25.66*</td>
</tr>
<tr>
<td>ComiRec-DR</td>
<td>FR</td>
<td>17.00</td>
<td>16.79</td>
<td>9.85</td>
<td>16.91</td>
<td>16.75</td>
<td>9.82</td>
<td>14.79</td>
<td>12.79</td>
<td>12.06</td>
<td>44.29</td>
<td>25.87</td>
</tr>
<tr>
<td></td>
<td>FT</td>
<td>15.41</td>
<td>15.35</td>
<td>-</td>
<td>15.36</td>
<td>15.28</td>
<td>-</td>
<td>13.30</td>
<td>11.30</td>
<td>-</td>
<td>42.62</td>
<td>24.68</td>
</tr>
<tr>
<td></td>
<td>SML</td>
<td>16.16</td>
<td>15.85</td>
<td>4.09</td>
<td>16.08</td>
<td>15.77</td>
<td>3.92</td>
<td>13.92</td>
<td>11.85</td>
<td>4.74</td>
<td>43.28</td>
<td>24.89</td>
</tr>
<tr>
<td></td>
<td>ADER</td>
<td>16.17</td>
<td>15.90</td>
<td>4.10</td>
<td>16.07</td>
<td>15.86</td>
<td>3.96</td>
<td>13.71</td>
<td>11.96</td>
<td>4.43</td>
<td>43.64</td>
<td>25.00</td>
</tr>
<tr>
<td></td>
<td>IMSR</td>
<td>16.80*</td>
<td>16.48*</td>
<td>8.20</td>
<td>16.74*</td>
<td>16.47*</td>
<td>8.38</td>
<td>14.46*</td>
<td>12.48*</td>
<td>9.51</td>
<td>44.48*</td>
<td>26.00*</td>
</tr>
<tr>
<td>ComiRec-SA</td>
<td>FR</td>
<td>17.15</td>
<td>16.95</td>
<td>10.82</td>
<td>16.74</td>
<td>16.87</td>
<td>8.83</td>
<td>14.86</td>
<td>12.85</td>
<td>11.66</td>
<td>44.31</td>
<td>25.75</td>
</tr>
<tr>
<td></td>
<td>FT</td>
<td>15.31</td>
<td>15.46</td>
<td>-</td>
<td>15.49</td>
<td>15.39</td>
<td>-</td>
<td>13.46</td>
<td>11.35</td>
<td>-</td>
<td>42.44</td>
<td>24.58</td>
</tr>
<tr>
<td></td>
<td>SML</td>
<td>15.96</td>
<td>15.99</td>
<td>3.33</td>
<td>15.90</td>
<td>15.88</td>
<td>2.89</td>
<td>13.78</td>
<td>11.71</td>
<td>2.72</td>
<td>43.17</td>
<td>24.83</td>
</tr>
<tr>
<td></td>
<td>ADER</td>
<td>16.32</td>
<td>15.88</td>
<td>4.63</td>
<td>16.14</td>
<td>15.88</td>
<td>3.67</td>
<td>13.55</td>
<td>11.08</td>
<td>2.87</td>
<td>43.43</td>
<td>25.00</td>
</tr>
</tbody>
</table>

IMSR achieves 3.77%, 3.89%, 4.21%, 4.76% relative improvements on NDCG compared to the second best incremental learning methods on four datasets.

IMSR is slightly inferior to FR by only using the newly collected interactions.
Effects of Different Modules in IMSR

All components in IMSR contribute to the recommendation performance.

The effectiveness of all three components is insensitive to the base model.

On Taobao, the effectiveness of the NID and PIT is more significant.

→ users in Taobao develop new interests fast due to the richness of item categories.

<table>
<thead>
<tr>
<th>dataset</th>
<th>$t_0$</th>
<th>$t_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>average interests number of all users</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Books</td>
<td>4</td>
<td>5.6</td>
</tr>
<tr>
<td>Taobao</td>
<td>4</td>
<td>9.2</td>
</tr>
</tbody>
</table>
Parameter Sensitivity

- Hyperparameters $c_1$ and $c_2$:
  - $c_1$ controls the new interests detecting sensitivity and $c_2$ controls the strictness of redundant interest trimming. The model achieves the highest performance with moderate values of $c_1$ and $c_2$ in most cases.

- Interests number $K$ and $\delta K$:
  - Best performance is achieved when $K=6$, $\delta K=3$ on Taobao, $K=4$, $\delta K=3$ on Books.
  - $K=19$, $\delta K=0$ (refer to creating all the interest vectors in advance) is far below the performance of IMSR with $K=4$, $\delta K=3$, which confirms using interests expansion strategy is more effective than creating too many interests in advance.
Interest Visualization

(b) t-SNE visualization of one user interest evolution among different time spans (colors).

Different interests are located in different places, which reflects the effectiveness of PIT and NID in preventing learning redundant interests.

Same interest in different time spans linked with red dashes locate closely, which shows that EIR prevents dramatical drift of existing interests.
Conclusions & Future work

- **New Framework IMSR** for incremental multi-interest sequential recommendation:
  - Retain and slightly drift existing interest vectors. → EIR
  - Detect the occurrence of new interests automatically. → NID
  - Expand new interests adaptively. → PIT

- **Prominent Performance** on two kinds of multi-interest sequential recommendation models
  - Apply IMSR on two kinds of MSR models: dynamic-routing-based and self-attention-based models.
  - Conduct extensive experiments on four real datasets where our proposed framework achieve superior performance compared with the existing incremental learning methods.

- **Future Work:**
  - How to compress user interest vectors under memory space limit?
  - How to remove interests being inactive for a long while?
Thank you!

- E-Mail: Cloudcatcher.888@sjtu.edu.cn
- Github: https://github.com/Cloudcatcher888/IMSR
- Homesite: https://cloudcatcher888.github.io

Q&A