Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation

SIGIR2020

PeterRec Data&Code: https://github.com/fajieyuan/sigir2020_peterrec

Fajie Yuan (Tencent); Xiangnan He (University of Science and Technology of China)
Alexandros Karatzoglou (Google Research); Liguang Zhang (Tencent)
Outline

➢ Motivation
➢ Related Work
➢ PeterRec
➢ Experiments
Users engage with recommender systems and provided or left feedback.
A user has different roles to play in life!
A user has different roles to play in life!
Our Motivation

A user has different roles to play in life!

My user Model
Outline

➢ Motivation
➢ Related Work
➢ PeterRec
➢ Experiments
• **Recommendation Background:**

(1) **Content & Context Recommendation**

(2) **Session-based Recommendation:** recommending the next item based on previously recorded user interactions.

A DSSM (Non-sequential) RS model (Supervised Learning)
Why sequential recommendation?

- Short-videos (TikTok, Weishi, Kuaishou)
- Music (Tencent music, Yahoo! Music) & News
- Movie clips (YouTube, Netflix)

NonSeq Rec vs. Seq Rec:

- Only Static vs. Dynamic Preference
- Manual Feature Engineering vs. Manual-free Features
- Supervised Learning vs. Unsupervised (self-supervised) Learning
Transfer Learning Background

TL aims to extract the knowledge from one or more source tasks and applies the knowledge to a target task.

Figure: A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning, online
Transfer Learning (TL) vs Multi-task Learning (MTL)

<table>
<thead>
<tr>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transfer Learning</strong></td>
<td><strong>Task 1</strong></td>
</tr>
<tr>
<td><strong>Multi-task Learning</strong></td>
<td><strong>Task 1</strong> ... <strong>Task N</strong></td>
</tr>
</tbody>
</table>

**TL vs MTL**
- Two-stage training vs joint training
- One objective vs multiple objectives
- Care only target vs. care all objectives

[1] ICML 2018: Advances in transfer, multitask, and semi-supervised learning, online
Transfer Learning (TL) for Recommender System (RS)

Motivation:
• User representation may be generic, since their preference tends to be similar across different recommendation task. That is, user’s engagement in previous platforms may be important training signals for other systems.
• Traditional ML models usually fail to when modeling new or cold users due to lack of interaction data.

[1] figure is from online, url is missing
Outline

➢ Motivation
➢ Related Work
➢ PeterRec
➢ Experiments
Task description

Source data: \((u, x^u),\) where \(x^u = \{x^u_1, x^u_2, \ldots, x^u_n\},\)

where \(x^u_t\) denotes the \(t\)-th interacted item of user \(u\)

Target data: \((u, y)\) where \(y\) is the supervise label in the target dataset

Example

Source data: user’s watching activities in Tencent QQ Browser

Target data: user’s watching activities in Kandian, but users are cold or new here

or user’s profile labels e.g. age, gender, lifestatus, etc.
PeterRec Architecture

NextItNet-style neural network

AutoRegressive

Label (e.g., gender)

user clicking behaviors

(a) pre-training on QQ Browser data

(b) fine-tuning on user profile dataset

What can be done by PeterRec

- Cold-start problem, e.g., ads rec
- User profile prediction, e.g., gender prediction

UserID: 007XX

Gender: Male
Age: 25-35
Profession: programmer
Life Status: unmarried
Health Status: healthy and positive
Tag: straight, beauty, Chinese, AI, Kung fu, sports
Training a separate model for each downstream task is parameter-inefficient since both pretraining & finetuning models are very large.

The number of finetuned models is as many as the number of downstream tasks. 100 tasks = 100 finetuned models.

Problems we meet when a number of tasks are required.
Taking inspiration from grafting

A: branch of plum  
B: Tree of peach  
C: insertion  
D: grow together

Pretrained model  
Pretrained model is treated as the peach Tree.
Grafting for plants.

A: branch of plum vs MP
B: Tree of peach vs pretrained model
C: insertion vs insertion
D: grow together vs finetuning

Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation, Yuan et al SIGIR 2020
Outline

➢ Motivation
➢ Related Work
➢ PeterRec
➢ Experiments
Results.

Is pretraining necessary?
PeterZero: no pretraining
PeterRec: with pretraining
Table 3: Performance comparison (with the non-causal CNN architectures). The number of fine-tuned parameters ($\vartheta$ and $\nu$) of PeterRec accounts for 9.4%, 2.7%, 0.16%, 0.16%, 0.16% of FineAll on the five datasets from left to right.

<table>
<thead>
<tr>
<th>Model</th>
<th>ColdRec-1</th>
<th>ColdRec-2</th>
<th>GenEst</th>
<th>AgeEst</th>
<th>LifeEst</th>
</tr>
</thead>
<tbody>
<tr>
<td>FineCLS</td>
<td>0.295</td>
<td>0.293</td>
<td>0.900</td>
<td>0.679</td>
<td>0.606</td>
</tr>
<tr>
<td>FineLast</td>
<td>0.330</td>
<td>0.310</td>
<td>0.902</td>
<td>0.682</td>
<td>0.608</td>
</tr>
<tr>
<td>FineAll</td>
<td>0.352</td>
<td>0.338</td>
<td>0.905</td>
<td>0.714</td>
<td>0.615</td>
</tr>
<tr>
<td>PeterRec</td>
<td>0.351</td>
<td>0.339</td>
<td>0.906</td>
<td>0.714</td>
<td>0.615</td>
</tr>
</tbody>
</table>
What can be done by PeterRec

Example: if we have the video watch behaviors of a teenager, we may know whether he has depression or propensity for violence by PeterRec without resorting to much feature engineering and human-labeled data.