Outer Product-based Neural Collaborative Filtering

Xiangnan He¹, Xiaoyu Du², Xiang Wang¹, Feng Tian³, Jinhui Tang⁴ and Tat-Seng Chua¹

1: National University of Singapore
2: Chengdu University of Information Technology
3: Northeast Petroleum University
4: Nanjing University of Science and Technology
A prevalent model for collaborative filtering

- Represent a user (or an item) as a vector of latent factors (also termed as *embedding*)
A prevalent model for collaborative filtering

- Represent a user (or an item) as a vector of latent factors (also termed as \textit{embedding})

- Estimate an interaction as the inner product between the user embedding and item embedding
Matrix Factorization (MF)

- A prevalent model for collaborative filtering
  - Represent a user (or an item) as a vector of latent factors (also termed as *embedding*)
  - Estimate an interaction as the inner product between the user embedding and item embedding

- Many extensions on MF
  - Model perspective: NeuMF [He et al, WWW’17], Factorization Machine etc.
  - Learning perspective: BPR, Adversarial Personalized Ranking [He et al, SIGIR’18]
MF uses **Inner Product** as the interaction function

The implicit assumption in Inner Product:

- The embedding dimensions are independent with each other

However, the implicit assumption is **impractical**.

- The embedding dimensions could be interpreted as certain properties of items [Zhang et al., SIGIR’14], which are **not necessarily** to be independent

Recent DNN-based models either use **element-wise product** or **concatenation**.

- E.g., NeuMF [He et al, WWW’17], NNCF [Bai et al, CIKM’17], JRL [Zhang et al, CIKM’17], Autoencoder-based CF Models [Wu et al, WSDM’16]

- Still, the relations among embedding dimensions are **not explicitly modeled**.
• How to model the relations between embedding dimensions?

• Next: our proposed method:
  1. Outer product on user & item embedding for pairwise interaction modeling
  2. CNN on the outer product matrix to extract and reweight prediction signals.
Outer-product \textbf{explicitly} models the \textbf{pairwise relations} between embedding dimensions:
- Get a 2D matrix, named as \textit{interaction map}:
  
  \[
  E = p_u \otimes q_i = p_u q_i^T = \\
  \begin{bmatrix}
  p_{u1} q_{i1} & p_{u1} q_{i2} & \ldots & p_{u1} q_{ik} \\
  p_{u2} q_{i1} & p_{u2} q_{i2} & \ldots & p_{u2} q_{ik} \\
  \vdots & \vdots & \ddots & \vdots \\
  p_{uk} q_{i1} & \ldots & \ldots & p_{uk} q_{ik}
  \end{bmatrix}
  \]

  Indicating the interaction between the \textit{k}-th dimension of \textit{p}_u and the 2-nd dimension of \textit{q}_i.

Signals in Inner product

© Copyright NExT++. All Right Reserved.
Above the interaction map are hidden layers, which aim to extract useful signal from the 2D interaction map.

A straightforward solution is to use MLP, however it results in too many parameters:

- Interaction map $E$ has $K \times K$ neurons ($K$ is embeddings size usually hundreds)
- Require large memories to store the model
- Require large training data to learn the model well
ConvNCF uses locally connected CNN as hidden layers in ONCF:

- CNN has much fewer parameters than MLP
- Hierarchical tower structure: higher layer integrates more information from larger area.
- Final prediction summarizes all information from interaction map.

- 2 Fully Connected Layers: > 10M parameters
- 6 Convolutional Layers: 20K parameters, but achieve better performance!
Datasets
- Yelp: 25,815 users, 25,677 items, and 730,791 interactions.
- Gowalla: 54,156 users, 52,400 items, and 1,249,703 interactions.

Protocols
- Leave-one-out: holdout the latest interaction of each user as the test
- Pair 1 testing instance with 999 negative instances
- Top-K evaluation: ranking 1 positive vs. 999 negatives.
- Ranking lists are evaluated by Hit Ratio and NDCG (@10).

Loss Function
- Bayesian Personalized Ranking
Baselines

- MF-BPR [Rendle et al., UAI’09]
  - Learning MF with a pair-wise classification loss.
- MLP [He et al., WWW’17]
  - 3-layer multi-layer perceptron above user and item embeddings.
- JRL [Zhang et al., CIKM’17]
  - Multi-layer perceptron above the element-wise product of embeddings.
- NeuMF [He et al., WWW’17]
  - A neural network combining hidden layer of MF and MLP.
## Performance Comparison

<table>
<thead>
<tr>
<th></th>
<th>Gowalla</th>
<th>Yelp</th>
<th>Average Improvement of ConvNCF over Baselines</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HR@10</td>
<td>NDCG@10</td>
<td>HR@10</td>
</tr>
<tr>
<td>MF-BPR</td>
<td>0.7480</td>
<td>0.5214</td>
<td>0.2817</td>
</tr>
<tr>
<td>MLP</td>
<td>0.7590</td>
<td>0.5202</td>
<td>0.2831</td>
</tr>
<tr>
<td>JRL</td>
<td>0.7747</td>
<td>0.5615</td>
<td>0.2922</td>
</tr>
<tr>
<td>NeuMF</td>
<td>0.7793</td>
<td>0.5660</td>
<td>0.2958</td>
</tr>
<tr>
<td>ConvNCF</td>
<td>0.7936*</td>
<td>0.5826*</td>
<td>0.3086*</td>
</tr>
</tbody>
</table>

* indicates that the improvements over all other methods are statistically significant for p < 0.05.

### Overall Performance:

- **ConvNCF > NeuMF** [He et al., 2017] > **JRL** [Zhang et al., 2017]

- Usefulness of modeling the relations of embedding dimensions
- Training MLP well is practically difficult.
Efficacy of Outer Product

Training process of neural models that apply different operations above the embedding layer:
- ConvNCF: outer product; GMF: element-wise product; MLP: concatenation; JRL: element-wise product

Outer product is a simple but effective merge of user & item embeddings.
NDCG@10 of using different hidden layers for ONCF:

- **ConvNCF** uses a 6-layer CNN.
- **ONCF-mlp** uses a 3-layer MLP above the interaction map.

1. ConvNCF outperforms ONCF-mlp.
2. ConvNCF is more stable than ONCF-mlp.
Conclusion & Future Work

Summary of contributions:

- A new neural framework for CF --- ONCF, which explicitly captures pairwise correlations between embedding dimensions with outer product.
- A new model of ONCF framework --- ConvNCF, which uses CNN as hidden layers.
- Extensive experiments show effectiveness of ONCF framework and ConvNCF method.

Future work:

- We will explore more advanced CNN models to further explore the potentials of our ONCF framework.
- We will extend ONCF to content-based recommendation scenarios, e.g., items have image and textual content.


THANK YOU!

Codes: https://github.com/duxy-me/ConvNCF

For questions: email to Dr. He Xiangnan
xiangnanhe@gmail.com