# Neural Collaborative Filtering

STAT3009 Recommender Systems

by Ben Dai (CUHK-STAT)
on October 9, 2024

# » Recall LFM: NCF

Recall the basic Latent Factor Model:

$$\min_{\boldsymbol{P},\boldsymbol{Q}} \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} (r_{ui} - \boldsymbol{p}_u^\intercal \boldsymbol{q}_i)^2 + \lambda \Big( \sum_{u=1}^{n} \|\boldsymbol{p}_u\|_2^2 + \sum_{i=1}^{n} \|\boldsymbol{q}_i\|_2^2 \Big) \tag{1}$$

* The **interaction** btw users and items are formulated as **inner production**.
* It can be extended to **high-order** nonlinear interaction.

## » Nonlinear interaction: NCF

* For a general nonlinear function $f$, the predicted rating can be formulated as,

$$\widehat{r}_{ui} = f(\boldsymbol{p}_u, \boldsymbol{q}_i).$$

* Nonlinear methods: **polynomials**, **B-splines**, **kernel methods**, and neural networks
* Now, we want to formulate the nonlinear RS as a neural network

# » NCF: Goal

Let's summarize our goal:

- LFM — The idea of "latent factor" is quite useful for RS
- NN — Attempt to incorporate "neural networks" to formulate **high-order** interaction between users and items.

Quick solution:

- Step 1 — Mapping $u \to \mathbf{p}_u$; and mapping $i \to \mathbf{q}_i$
- Step 2 — Define a network $f : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$
- Step 3 — The final prediction is $\widehat{r}_{ui} = f(\mathbf{p}_u, \mathbf{q}_i)$

How to estimate latent factors $(\mathbf{P}, \mathbf{Q})$ and the network $f$?

## » NCF: Separate formulation

The neural network based on LFM can be formulated as:

$$\underset{\mathbf{P},\mathbf{Q},f}{\operatorname{argmin}} \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} (r_{ui} - f(\boldsymbol{p}, \boldsymbol{q}_i))^2 + \lambda \Big( \sum_{u=1}^{n} \|\boldsymbol{p}_u\|_2^2 + \sum_{i=1}^{n} \|\boldsymbol{q}_i\|_2^2 \Big)$$

## » NCF: Separate formulation

The neural network based on LFM can be formulated as:

$$\underset{\mathbf{P},\mathbf{Q},f}{\operatorname{argmin}} \ \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} (r_{ui} - f(\boldsymbol{p}, \boldsymbol{q}_i))^2 + \lambda \left( \sum_{u=1}^{n} \|\boldsymbol{p}_u\|_2^2 + \sum_{i=1}^{n} \|\boldsymbol{q}_i\|_2^2 \right)$$

**Opt**: B-SGD

- P Update **P** by fixing **Q** and $f$
- Q Update **Q** by fixing **P** and $f$
- f Update $f$ by fixing **P** and **Q**

**A bit cumbersome** to implement

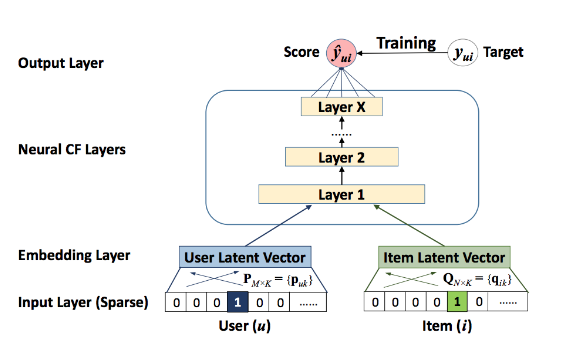
* Auto-gradient in TensorFlow
* Can we incorporate latent factors **P** and **Q** as params of a unified neural network
* Then just SGD update *all* **params** in the network

## » NCF: Unified formulation

A unified neural network

Input user-item pair $(u, i) \in \{0, \cdots, n\} \times \{0, \cdots, m\}$

Output the predicted rating




Source[1]

[1] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T. S. (2017, April). Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web (pp. 173-182).

## » NCF: embedding layer

We can incorporate **P** and **Q** by introducing embedding layers in the network;

A embedding layer attempt to

One-hot Encoding **userId** (**itemId**) as one-hot encoding (a binary dummy vector)

$$u \rightarrow \mathbf{e}_{n,u} = (0, \cdots, \underbrace{1}_{u.th}, \cdots, 0)^\top, \quad i \rightarrow \mathbf{e}_{m,i} = (0, \cdots, \underbrace{1}_{i.th}, \cdots, 0)^\top$$

Product Product one-hot encoding with the embedding matrix

$$\mathbf{p}_u = \mathbf{e}_{n,u}^\top \mathbf{P}, \quad \mathbf{q}_i = \mathbf{e}_{m,i}^\top \mathbf{Q}$$

Indeed, embedding is essentially same with **latent factors**; just different names...

## » NCF: embedding layer

Let's summarize Embedding Layer...

Mapping $u \to \mathbf{p}_u$, or cate_feat $\to$ dense representation

Params Embedding matrix: #User (or #Item) $\times$ #LatentFactor

hp #LatentFactor or embedding size

Opt The model becomes

$$\widehat{f_\theta} = \underset{f_\theta}{\operatorname{argmin}} \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} (r_{ui} - f_{\boldsymbol{\theta}}(u,i))^2 + \lambda \mathrm{Reg}(\boldsymbol{\theta})$$

## » NCF: RS formulation

Model The model for NCF is:

$$\widehat{r}_{ui} = f([\mathbf{a}_u, \mathbf{b}_i])$$

Params All params in the network: two embedding matrices in embedding layers and weights in the other layers

hp embedding size and network architecture

Data $(u, i) \to r_{ui}$

Loss MSE

Metrics RMSE, MAE, ...

Opt SGD/B-SGD

**InClass demo: Neural collaborative filtering #1**

# » ANCF: Additional formulation

Model We want to keep the LFM terms in the model...

$$\widehat{r}_{ui} = \mathbf{p}_u^\intercal \mathbf{q}_i + f([\mathbf{a}_u, \mathbf{b}_i])$$

Params All params in the network: two embedding matrices in embedding layers and weights in the other layers

hp embedding size and network architecture

Data $(u, i) \to r_{ui}$

Loss MSE

Metrics RMSE, MAE, ...

Opt SGD/B-SGD

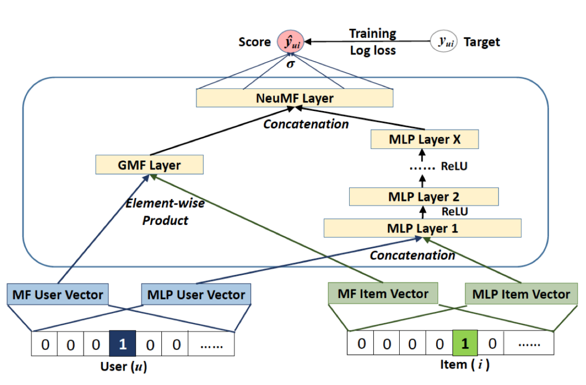
**InClass demo: Neural collaborative filtering #2**

# » Neural-NCF: neural formulation

More general operators based on LFM terms

$$\hat{r}_{ui} = f_1\left([\mathbf{p}_u \circ \mathbf{q}_i, f_2(\mathbf{a}_u^\top, \mathbf{b}_i)]\right)$$

input and output remain the same



2

2 He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T. S. (2017, April). Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web (pp. 173-182).

## » NCF: RS formulation

Model The model becomes

$$\widehat{r}_{ui} = f_1\left([\mathbf{p}_u \circ \mathbf{q}_i, f_2(\mathbf{a}_u^\intercal, \mathbf{b}_i)]\right)$$

Params All params in the network: four embedding matrices in embedding layers and weights in the other layers

hp embedding size and network architecture

Data $(u, i) \to r_{ui}$

Loss MSE

Metrics RMSE, MAE, ...

Opt SGD/B-SGD

**InClass demo: Neural collaborative filtering #3**