

# Northumbria Research Link

Citation: Jiao, Lihong, Yu, Yonghong, Zhou, Ningning, Zhang, Li and Yin, Hongzhi (2020) Neural Pairwise Ranking Factorization Machine for Item Recommendation. In: Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24-27, 2020, Proceedings, Part I. Lecture Notes in Computer Science (12112). Springer, Cham, pp. 680-688. ISBN 9783030594091, 9783030594107

Published by: Springer

URL: [https://doi.org/10.1007/978-3-030-59410-7\\_46](https://doi.org/10.1007/978-3-030-59410-7_46) <[https://doi.org/10.1007/978-3-030-59410-7\\_46](https://doi.org/10.1007/978-3-030-59410-7_46)>

This version was downloaded from Northumbria Research Link: <https://nrl.northumbria.ac.uk/id/eprint/44485/>

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: <http://nrl.northumbria.ac.uk/policies.html>

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)

# Neural Pairwise Ranking Factorization Machine for Item Recommendation

Lihong Jiao<sup>1</sup>, Yonghong Yu<sup>2</sup>, Ningning Zhou<sup>1</sup>, Li Zhang<sup>3</sup>, and Hongzhi Yin<sup>4</sup>

<sup>1</sup> School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. zhounn@njupt.edu.cn

<sup>2</sup> Tongda College, Nanjing University of Posts and Telecommunications, Nanjing, China. yuyh@njupt.edu.cn

<sup>3</sup> Department of Computer and Information Sciences, Northumbria University, Newcastle, UK. li.zhang@northumbria.ac.uk

<sup>4</sup> School of Information Technology and Electrical Engineering, The University of Queensland, Australia. db.hongzhi@gmail.com

**Abstract.** The factorization machine models attract significant attention from academia and industry because they can model the context information and improve the performance of recommendation. However, traditional factorization machine models generally adopt the point-wise learning method to learn the model parameters as well as only model the linear interactions between features. They fail to capture the complex interactions among features, which degrades the performance of factorization machine models. In this paper, we propose a neural pairwise ranking factorization machine for item recommendation, which integrates the multi-layer perceptual neural networks into the pairwise ranking factorization machine model. Specifically, to capture the high-order and nonlinear interactions among features, we stack a multi-layer perceptual neural network over the bi-interaction layer, which encodes the second-order interactions between features. Moreover, the pair-wise ranking model is adopted to learn the relative preferences of users rather than predict the absolute scores. Experimental results on real world datasets show that our proposed neural pairwise ranking factorization machine outperforms the traditional factorization machine models.

**Keywords:** Recommendation Algorithm, Factorization Machine, Neural Networks

## 1 Introduction

With the development of information technology, a variety of network applications have accumulated a huge amount of data. Although the massive data provides users with rich information, it leads to the problem of “information overload”. The recommendation systems [1] can greatly alleviate the problem of information overload. They infer users latent preferences by analyzing their past activities and provide users with personalized recommendation services. Factorization machine (FM) [2] model is very popular in the field of recommendation

systems, which is a general predictor that can be adopted for the prediction tasks working with any real valued feature vector.

Recently, deep learning techniques have shown great potential in the field of recommendation systems, some researchers also have utilized deep learning techniques to improve the classic factorization machine models. Typical deep learning based factorization machine models include NFM [3], AFM [4], and CFM [5]. Neural Factorization Machine (NFM) [3] seamlessly unifies the advantages of neural networks and factorization machine. It not only captures the linear interactions between feature representations of variables, but also models nonlinear high-order interactions. However, both FM and NFM adopt a point-wise method to learn their model parameters. They fit the user’s scores rather than learn the user’s relative preferences for item pairs. In fact, common users usually care about the ranking of item pairs rather than the absolute rating on each item. Pairwise ranking factorization machine (PRFM) [6, 7] makes use of the bayesian personalized ranking (BPR) [8] and FM to learn the relative preferences of users over item pairs. Similar to FM, PRFM can only model the linear interactions among features.

In this paper, we propose the Neural Pairwise Ranking Factorization Machine (NPRFM) model, which integrates the multi-layer perceptual neural networks into the PRFM model to boost the recommendation performance. Specifically, to capture the high-order and nonlinear interactions among features, we stack a multi-layer perceptual neural network over the bi-interaction layer, which is a pooling layer that encodes the seconde-order interactions between features. Moreover, the bayesian personalized ranking criterion is adopted to learn the relative preferences of users, which makes non-observed feedback contribute to the inference of model parameters. Experimental results on real world datasets show that our proposed neural pairwise ranking factorization machine model outperforms the traditional recommendation algorithms.

## 2 Preliminaries

Factorization Machine is able to model the interactions among different features by using a factorization model. Usually, the model equation of FM is defined as follows:

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \quad (1)$$

where  $\hat{y}(\mathbf{x})$  is the predicted value, and  $\mathbf{x} \in R^n$  denotes the input vector of the model equation.  $x_i$  represents the  $i$ -th element of  $\mathbf{x}$ .  $w_0 \in R$  is the global bias,  $\mathbf{w} \in R^n$  indicates the weight vector of the input vector  $\mathbf{x}$ .  $\mathbf{V} \in R^{n \times k}$  is the latent feature matrix, whose  $\mathbf{v}_i$  represents the feature vector of  $x_i$ .  $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$  is the dot product of two feature vectors, which is used to model the interaction between  $x_i$  and  $x_j$ .

### 3 Neural Pairwise Ranking Factorization Machine

To model the high-order interaction behaviors among features as well as learn the relatively preferences of user over item pairs, we propose the neural pairwise ranking factorization machine (NPRFM) model, whose underlying components are NFM and PRFM. Fig. 1 presents the framework of our proposed neural pairwise ranking factorization machine, which consists of four layers, i.e. embedding layer, Bi-interaction layer, hidden layer and prediction layer.

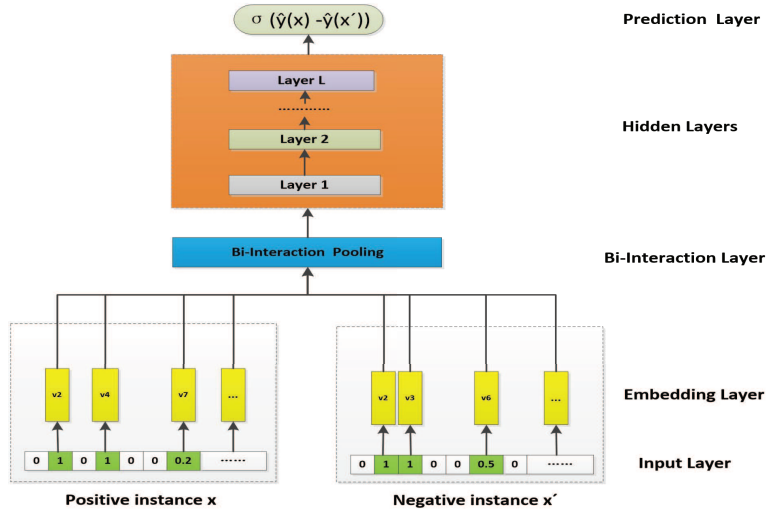


Fig. 1. The framework of neural pairwise ranking factorization machine

The input of NPRFM includes positive and negative instances. Both positive or negative instance contain user, item and context information. By using one-hot encoding, the positive and negative instances are converted into sparse feature vectors  $\mathbf{x} \in R^n$  or  $\mathbf{x}' \in R^n$ , respectively.

#### 3.1 Embedding Layer

After one-hot encoding, we use the embedding table lookup operation to obtain the embedded representations of features included in the input instance. Formally, the embedded representation of  $\mathbf{x}$  is,

$$\mathbf{V}_x = \mathbf{V}.\text{onehot}(\mathbf{x}) \quad (2)$$

where  $\mathbf{V}_x$  is a set of embedding vectors, i.e.,  $\mathbf{V}_x = \{x_1 \mathbf{v}_1, \dots, x_n \mathbf{v}_n\}$ , and  $\mathbf{v}_i \in R^k$  is the embedded representation of the  $i$ -th feature.

### 3.2 Bi-Interaction Layer

The Bi-Interaction layer is a pooling operation, which converts the set of embedding vectors  $\mathbf{V}_x$  to one vector  $f_{BI}(\mathbf{V}_x)$ :

$$f_{BI}(\mathbf{V}_x) = \sum_{i=1}^n \sum_{j=i+1}^n x_i \mathbf{v}_i \odot x_j \mathbf{v}_j \quad (3)$$

where  $\odot$  represents the element-wise product of two vectors. As shown in Eq.(3), the Bi-interaction layer captures the pair-wise interactions among the low dimensional representations of features. In other words, the Bi-Interaction pooling only encodes the second-order interactions among features.

### 3.3 Hidden Layers and Prediction Layer

Since the Bi-interaction layer only captures the second-order interactions among features, and can not model the complexity interactive patterns among features, we utilize the multi-layer perceptron (MLP) to learn the interaction relationships among features, which endows our proposed model with the ability of capturing the high-order interactions. Specifically, in the hidden layers, we stack multiple fully connected hidden layers over the Bi-interaction layer, where the output of a hidden layer is fed into the following hidden layer that makes use of the weighted matrix and non-linear activation function, such as sigmoid, tanh and ReLU, to nonlinearly transform this output. Formally, the MLP is defined as,

$$\begin{aligned} \mathbf{z}_1 &= \sigma_1 (\mathbf{W}_1 f_{BI}(\mathbf{V}_x) + \mathbf{b}_1), \\ \mathbf{z}_2 &= \sigma_2 (\mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2), \\ &\dots \\ \mathbf{z}_L &= \sigma_L (\mathbf{W}_L \mathbf{z}_{L-1} + \mathbf{b}_L) \end{aligned} \quad (4)$$

where  $L$  denotes the number of hidden layers,  $\mathbf{W}_l$ ,  $\mathbf{b}_l$  and  $\sigma_l$  represent the weight matrix, bias vector and activation function for the  $l$ -th layer, respectively.

The prediction layer is connected to the last hidden layer, and is used to predict the score  $\hat{y}(\mathbf{x})$  for the instance  $\mathbf{x}$ , where  $\mathbf{x}$  can be positive or negative instances. Formally,

$$\hat{y}(\mathbf{x}) = \mathbf{h}^T \mathbf{z}_L \quad (5)$$

where  $\mathbf{h}$  is the weight vector of the prediction layer. Combining the Eq. (4) and (5), the model equation of NPFFM is reformulated as:

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^n w_i x_i + \mathbf{h}^T \sigma_L (\mathbf{W}_L (\dots \sigma_1 (\mathbf{W}_1 f_{BI}(\mathbf{V}_x) + \mathbf{b}_1) \dots) + \mathbf{b}_L) \quad (6)$$

### 3.4 Model Learning

We adopt a ranking criterion, i.e. the Bayesian personalized ranking, to optimize the model parameters of NPRFM. Formally, the objective function of NPRFM is defined as:

$$\mathcal{L}^{NPRFM} = \sum_{(\mathbf{x}, \mathbf{x}') \in \chi} -\ln \sigma(\hat{y}(\mathbf{x}) - \hat{y}(\mathbf{x}')) + \frac{\lambda}{2} (\|\Theta\|_F^2) \quad (7)$$

where  $\sigma(\cdot)$  is the logistic sigmoid function. And  $\Theta = \{w_i, \mathbf{W}_l, \mathbf{b}_l, \mathbf{v}_i, \mathbf{h}\}$ ,  $i \in (1 \dots n)$ ,  $l \in (1 \dots L)$  denotes the model parameters.  $\chi$  is the set of positive and negative instances. We adopt the Adagrad [9] optimizer to update model parameter because the Adagrad optimizer utilizes the information of the sparse gradient and gains an adaptive learning rate, which is suitable for the scenarios of data sparse.

## 4 Experiments

### 4.1 DataSets and Evaluation Metrics

In our experiments, we choose two real-world implicit feedback datasets: Frappe<sup>5</sup> and Last.fm<sup>6</sup>, to evaluate the effectiveness of our proposed model.

The **Frappe** contains 96,203 application usage logs with different contexts. Besides the user ID and application ID, each log contains eight contexts, such as weather, city and country and so on. We use one-hot encoding to convert each log into one feature vector, resulting in 5382 features.

The **Last.fm** was collected by Xin et al. [5]. The contexts of user consist of the user ID and the last music ID listened by the specific user within 90 minutes. The contexts of item include the music ID and artist ID. This dataset contains 214,574 music listening logs. After transforming each log by using one-hot encoding, we get 37,358 features.

We adopt the leave-one-out validation to evaluate the performance of all compared methods, which has been widely used in the literature [4, 5, 10]. In addition, we utilize two widely used ranking based metrics, i.e., the Hit Ratio (*HR*) and Normalized Discounted Cumulative Gain (*NDCG*), to evaluate the performance of all comparisons.

### 4.2 Experimental Settings

In order to evaluate the effectiveness of our proposed algorithm, we choose FM, NFM, PRFM as baselines.

- **FM** : FM [2, 11] is a strong competitor in the field of context-aware recommendation, and captures the interactions between different features by using a factorization model.

<sup>5</sup> <http://baltrunas.info/research-menu/frappe>

<sup>6</sup> <http://www.dtic.upf.edu/ocelma/MusicRecommendationDataset>

- **NFM** : NFM [10] seamlessly integrates neural networks into factorization machine model. Based on the neural networks, NFM can model nonlinear and high-order interactions between latent representations of features.
- **PRFM**: PRFM [7] applies the BPR standard to optimize its model parameters. Different from FM and NFM, PRFM focuses on the ranking task that learns the relative preferences of users for item pairs rather than predicts the absolute ratings.

For all compared methods, we set dimension of the hidden feature vector  $k = 64$ . In addition, for FM, we set the regularization term  $\lambda = 0.01$  and the learning rate  $\eta = 0.001$ . For NFM, we set the number of hidden layers is 1, the regularization term  $\lambda = 0.01$  and the learning rate  $\eta = 0.001$ . For PRFM, we set the regularization term  $\lambda = 0.001$  and the learning rate  $\eta = 0.1$ . For NPRFM, we set the regularization term  $\lambda = 0.001$ , the learning rate  $\eta = 0.1$ , and the number of hidden layers  $L = 1$ . In addition, we initialize the latent feature matrix  $V$  of NPRFM with the embedded representations learned by PRFM.

### 4.3 Performance Comparison

We set the length of recommendation list  $n = 3, 5, 7$  to evaluate the performance of all compared methods. The experimental results on the two datasets are shown in Tables 1 and 2.

**Table 1.** Performance comparison on the Frappe dataset ( $k=64$ )

Recommendation Algorithm	$n=3$		$n=5$		$n=7$	
	HR	NDCG	HR	NDCG	HR	NDCG
FM	0.2445	0.1795	0.3050	0.2107	0.3422	0.2216
NFM	0.2510	0.1797	0.3702	0.2199	0.4686	0.2504
PRFM	0.4650	0.3868	0.5654	0.4280	0.6383	0.4533
NPRFM	0.4786	0.3962	0.5751	0.4358	0.6469	0.4607

**Table 2.** Performance comparison on the Last.fm dataset ( $k=64$ )

Recommendation Algorithm	$n=3$		$n=5$		$n=7$	
	HR	NDCG	HR	NDCG	HR	NDCG
FM	0.0770	0.0584	0.1064	0.0706	0.1344	0.0803
NFM	0.0972	0.0723	0.1372	0.0886	0.1702	0.1000
PRFM	0.1828	0.1374	0.2545	0.1667	0.3094	0.1857
NPRFM	0.1855	0.1402	0.2624	0.1715	0.3219	0.1921

From Tables 1 and 2, we have the following observations: (1) On both datasets, FM performs the worst among all the compared methods. The reason is that FM learns its model parameters by adopting a point-wise learning scheme, which usually suffers from data sparsity. (2) NFM is superior to FM with regards to all

evaluation metrics. One reason is that the non-linear and high-order interactions among representations of features are captured by utilizing the neural networks, resulting in the improvement of recommendation performance. (3) On both datasets, PRFM achieves better performance than those of FM and NFM. This is because PRFM learns its model parameters by applying the BPR criterion, in which the pair-wise learning method is used to infer the latent representations of users and items. To some extent, the pair-wise learning scheme is able to alleviate the problem of data sparsity by making non-observed feedback contribute to the learning of model parameters. (4) Our proposed NPRFM model consistently outperforms other compared methods, which demonstrates the effectiveness of the proposed strategies. When  $n = 3$ , NPRFM improves the  $HR$  of PRFM by 2.9% and 1.5% on Frappe and Last.fm, respectively. In terms of  $NDCG$ , the improvements of NPRFM over PRFM are 2.4% and 2.0% on Frappe and Last.fm, respectively.

#### 4.4 Sensitivity Analysis

**Impact of the depth of neural networks** In this section, we conduct a group of experiments to investigate the impact of the depth of neural networks on the recommendation quality. We set  $n = 5$ ,  $k = 64$ , and vary the depth of neural networks from 1 to 3.

In Table 3, NPRFM- $i$  denotes that NPRFM model with  $i$  hidden layers, especially, NPRFM-0 is equal to PRFM. We only present the experimental results on  $HR@5$  in Table 3 and the experimental results on  $NDCG@5$  show similar trends.

**Table 3.** Impact of  $L$

Methods	Frappe	Lastfm
NPRFM-0	0.5654	0.2545
NPRFM-1	0.5751	0.2624
NPRFM-2	0.5592	0.2572
NPRFM-3	0.5654	0.2077

**Table 4.** Impact of  $k$

$k$	Frappe	Lastfm
16	0.4650	0.1641
32	0.5515	0.2027
64	0.5751	0.2624
128	0.5692	0.2514

From Table 3, we observe that NPRFM has the best performance when the number of the hidden layer is equal to one, and the performance of NPRFM degrades when the number of the hidden layer increases. This is because the available training data is not sufficient for NPRFM to accurately learn its model parameters when the number of hidden layers is relatively large. By contrast, if the number of layers is small, NPRFM has limited ability of modeling the complex interactions among embedded representations of features, resulting in the sub-optimal recommendation performance.

**Impact of  $k$**  In this section, we conduct another group of experiments to investigate the impact of the dimension of embedded representations of features  $k$  on the recommendation quality.

As shown in Table 4, our proposed NPRFM model is sensitive to the dimension of embedded representation of feature. We find that the performance of



NPFFM is optimal when the dimension of embedded representation of feature is equal to 64. A possible reason is that the proposed model already has enough expressiveness to describe the latent preferences of user and characteristics of items when  $k = 64$ .

## 5 Conclusion

In this paper, we propose the neural pairwise ranking factorization machine model, which integrates the multi-layer perceptual neural networks into the PRFM model to boost the recommendation performance of factorization model. Experimental results on real world datasets show that our proposed neural pairwise ranking factorization machine model outperforms the traditional recommendation algorithms.

**Acknowledgments.** This work is supported in part by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (Grant No. 17KJB520028 ), NUPTSF (Grant No. NY217114), Tongda College of Nanjing University of Posts and Telecommunications (Grant No. XK203XZ18002) and Qing Lan Project of Jiangsu Province.

## References

1. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, 2005.
2. Steffen Rendle. Factorization machines. In *ICDM*, pages 995–1000. IEEE, 2010.
3. Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *SIGIR*, pages 355–364. ACM, 2017.
4. Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617*, 2017.
5. Xin Xin, Bo Chen, Xiangnan He, Dong Wang, Yue Ding, and Joemon Jose. Cfm: convolutional factorization machines for context-aware recommendation. In *IJCAI*, pages 3926–3932. AAAI Press, 2019.
6. Fajie Yuan, Guibing Guo, Joemon M Jose, Long Chen, Haitao Yu, and Weinan Zhang. Lambdafm: learning optimal ranking with factorization machines using lambda surrogates. In *CIKM*, pages 227–236. ACM, 2016.
7. Weiyu Guo, Shu Wu, Liang Wang, and Tieniu Tan. Personalized ranking with pairwise factorization machines. *Neurocomputing*, 214:191–200, 2016.
8. Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.
9. You Lu, Jeffrey Lund, and Jordan Boyd-Graber. Why adagrad fails for online topic modeling. In *EMNLP*, pages 446–451, 2017.
10. Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.
11. Steffen Rendle. Factorization machines with libfm. *TIST*, 3(3):57, 2012.