Information Processing and Management 61 (2024) 103525

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

# GPR-OPT: A Practical Gaussian optimization criterion for implicit recommender systems☆

Ting Bai [a], Xiaotong Wang [a], Zhenhao Zhang [a], Wei Song [a], Bin Wu [a,*], Jian-Yun Nie [b]

[a] Beijing University of Posts and Telecommunications, Xitu Cheng Road, Beijing, 100876, China
[b] University of Montreal, Montreal, Canada

ARTICLE INFO

*Keywords:*
Gaussian optimization
Recommender systems
Implicit feedback
Preference confidence

ABSTRACT

Implicit recommendation refers to the users' feedback on items derived from their interactions with items, i.e., clicks, and purchases. The methods in the implicit recommendation scenario usually regard all the adopted items as their favorites and indiscriminately assign a uniform confidence weight of their preference toward all the adopted items. In practice, however, a user's preferences toward different items vary a lot. Treating them equally in existing implicit feedback recommender systems may limit the capacity of learning algorithms. To address this problem, we propose a novel Gaussian Personalized Recommendation OPTimization criterion (GPR-OPT), and our aim is to make the unknown preference confidence of users toward their adopted items in implicit feedback recommendation be learnable, so as to improve the accuracy of implicit recommender systems. In particular, we assume the user's interests in items follow Gaussian distributions. By maximizing the posterior probability of items derived from the Gaussian distribution of user features, GPR-OPT is able to self-adaptively learn the confidence of users' preferences from the implicit user–item interactions. We conduct extensive experiments on three real-world datasets, i.e., Movielens 1M, Amazon Book, and Yelp, which show an average of 11.64% improvements over different kinds of collaborative filtering algorithms. GPR-OPT is a generic optimization criterion and can be easily integrated into most existing collaborative filtering recommendation models, leading to a great impact on implicit recommender systems.

## 1. Introduction

With the rapid increase in the volume of information on the Internet, recommender systems play important roles in people's daily lives by filtering out useful information to users. By displaying products to users who have interests in them, recommender systems create commercial value for the companies and make huge profits for various e-commercial platforms, e.g., the Amazon platform,[1] and the JD platform.[2] Increasing attention has been paid to the research of recommender systems in recent years. It has been widely studied in many domains, like e-commerce, news portals, video portals, and so on. There are two types of recommendation scenarios in recommender systems according to the collected information from users: explicit feedback and implicit feedback. In explicit recommendation, the preferences of users are explicitly expressed, for example, users give ratings (from 1 to 5) to items according

☆ This document is the results of the research project funded by the National Natural Science Foundation of China under Grant No. 62102038.
* Corresponding author.
*E-mail address:* wubin@bupt.edu.cn (B. Wu).
1 https://www.amazon.com/.
2 https://www.jd.com/?country=USA.

https://doi.org/10.1016/j.ipm.2023.103525
Received 24 April 2023; Received in revised form 24 September 2023; Accepted 6 October 2023
Available online 17 October 2023
0306-4573/

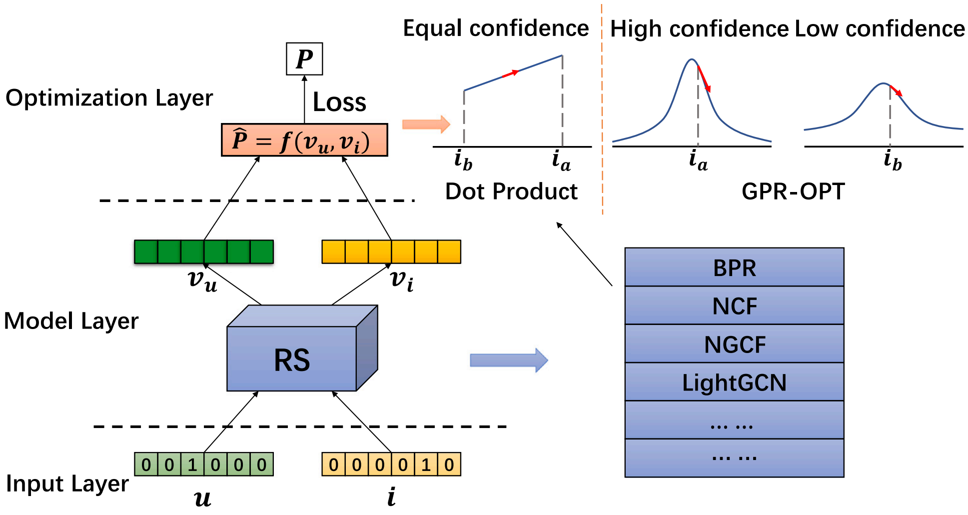



**Fig. 1.** The Generic Framework of Recommender Systems. $\mathbf{v}_\mathbf{u}$ and $\mathbf{v}_i$ are the learned vector of the user and item in the existing recommendation model, $f$ is the interaction function of two vectors for the prediction result $\hat{p}$. For different items $i_a$ and $i_b$, equal confidence (gradient) is assigned in the existing RS algorithms with dot product similarity function, while different preference confidences are self-adaptively learned in GPR-OPT. The red arrow represents the gradient in the loss function. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to their preference degree. However, in most real-world recommender systems, most feedback is not explicit but implicit. The explicit preference of users is always incomplete due to individual privacy issues or the high cost of human resources to collect them. More available is the implicit feedback, for example, the click, purchase, and viewing records of users in log files, which is easier to track and is already available in almost all information systems. Different from explicit recommendation, in which users explicitly show their preference degree, in the recommender systems with implicit feedback, we cannot directly map the interactions (e.g., click, purchase) to a user's preference due to the unknowable attitude of the user toward the adopted items (Parra, Karatzoglou, Amatriain, & Yavuz, 2011). Because even if a user clicks an item, he or she may not like it actually. Most existing Collaborative Filtering (CF) methods, for example, the typical matrix factorization-based methods (Mnih & Salakhutdinov, 2008; Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2009) and neural graph-based collaborative filtering methods (He et al., 2020; Wang, He, Wang, Feng, & Chua, 2019) usually adopt dot product as the similarity (or interaction) function, they treat all interacted items as user's favorite taste and assign a uniform confidence weight to learn from all the positive user–item interactions (as shown in Fig. 1). However, such a uniform assumption of user interests is invalid in real-world scenarios. A user may have different preferences toward different items. It is important to cope with this fact in order to improve the accuracy of existing recommender systems.

Some efforts had been made in previous studies to incorporate different preference confidences of users towards different items. They can be generally classified into three types: (1) using parameters to control the confidence levels of users' preferences towards items. For example, the confidence parameter is assigned according to the frequency of the interactions with items (Hu, Koren, & Volinsky, 2008; Liang, Charlin, McInerney, & Blei, 2016), and is learned to capture the uncertainty of the representations of users/items (Dos Santos, Piwowarski, & Gallinari, 2017; Jiang, Yang, Xiao, & Shen, 2019); (2) adopting neural networks, *e.g.,* MLP-based learned function, to automatically learn it from input features (Chen et al., 2020; He et al., 2017). For example, NCF (He et al., 2017) adopts a neural network-based similarity function to learn the different preferences of users. However, it has been proved in Rendle, Krichene, Zhang, and Anderson (2020) that the model with MLP-based learned similarity is less effective than dot product similarity. Distinguishing preference confidence of users can be learned in these models, but still in an inexplicable way, that is to say, they cannot explicitly learn the preference of users in implicit feedback recommender systems, leading to poor interpretability of the model. One can solve this problem by (3) mapping implicit feedback to explicit one via a logistic regression model (Parra et al., 2011) or using extra information, *e.g.,* review text or unclick behavior, to learn the explicit feedback (Jadidinejad, Macdonald, & Ounis, 2019; Xie et al., 2020). However, they still need to be supervised by the extra explicit feedback information.

To address this problem, we propose a novel probability model, **G**aussian **P**ersonalized **R**ecommendation **OPT**imization criterion (GPR-OPT), to automatically learn the real preference degree (termed as *"Preference Confidence"*) of a user toward his adopted items from implicit feedback. We explicitly characterize the preference confidence of a user as a learnable variable and assume it follows the Gaussian distribution,[3] which had been proved widely exists in most cases of our life according to Central Limit Theorem (Rosenblatt, 1956). As shown in Fig. 1, different from the dot product similarity used in the existing CF recommendation algorithms, in which different (positive) items $i_a$ and $i_b$ are assigned equal preference confidences (*i.e.,* the same importance), in GPR-OPT, the implicit feedback is learnable: the smaller the standard deviation of the Gaussian distribution, the higher the preference

[3] Considering the users' preference toward items may vary with the individuals and is influenced by many unknown random variables, we assume the preference confidence of a user toward items follows a normal distribution (*i.e.,* Gaussian distribution) according to Central Limit Theorem.

confidence, the more a user likes the item, and our Gaussian-based optimization will return larger gradients to punish the prediction bias.

GPR-OPT pays more attention to the interactions with higher confidence of user preference. It converts implicit feedback into a learnable explicit RS model without any extra information and offers a novel way to give explanations to recommendation results according to the explicit preference confidence learned by the model. Besides, It is a generic optimization criterion for implicit RS, which can be flexibly added on top of any collaborative filtering recommendation models. We found few research works had addressed the problem of making the implicit feedback learnable to improve the existing implicit RS algorithms. Our contributions are summarized as follows:

- We propose a novel Gaussian-based optimization criterion GPR-OPT for implicit recommender systems. It converts the implicit feedback to learnable explicit feedback with self-adaptive confidence.
- GPR-OPT is a light optimization criterion and can be easily adapted into most collaborative filtering recommendation models, which may bring a broader impact for improving existing recommender systems.
- We conduct extensive experiments on three representative datasets (Movielens 1M, Amazon Book, and Yelp), demonstrating the effectiveness of our optimization criterion for implicit collaborative filtering recommender systems.

## 2. Preliminary

### 2.1. The definition of implicit RS

Implicit recommendation refers to that users' feedback on items derived from users' interactions with items, i.e., clicks, and purchases. The implicit recommendation problem can be defined as follows: Given a user $u \in U$, and item $i \in I$, where $U$ and $I$ are the sets of users and items, the purchase records of $u$ is $I^u = \{i_1, i_2, \dots, i_n\}$, sorted by the corresponding timestamp of items. For a user $u$, and item $i$, the recommender system aims to predict the preference of $u$ toward $i$ by:

$$\hat{p}(u,i) = \mathcal{F}(i \in I | u, I^u), \tag{1}$$

where $\hat{p}(u,i)$ is the probability of item $i \in I$ being purchased by $u$ at the next time, and $\mathcal{F}$ is the prediction function. By ranking all the items according to the predicted probability, we recommend the top $K$ items to the user.

### 2.2. Generic implicit RS framework

As shown in Fig. 1, the input of RS is the interaction pair $\langle u, i \rangle$ and its context information, denoted as $\mathbf{x}_u$ and $\mathbf{x}_i$ respectively. By using any model of RS, the corresponding representations $\mathbf{v}_u$ and $\mathbf{v}_i$ can be learned. Generally, the prediction function is defined as:

$$\hat{p}(u,i) = \mathcal{F}(\mathbf{v}_u, \mathbf{v}_i) = \mathbf{v}_u \cdot \mathbf{v}_i. \tag{2}$$

Here we only consider the most widely used function, *i.e.,* the dot product similarity function, in RS models, which had been proved to be more effective than the MLP similarity function in RS (Rendle et al., 2020).

One obvious shortcoming of the above prediction function is that: it assigns a uniform weight to all the different items. For example, for two items $i_a$ and $i_b$ with different preference of a user $u$, assuming $\hat{p}(u, i_a) > \hat{p}(u, i_b)$ (see in Fig. 1), the RS model is optimized by:

$$\frac{\partial \hat{p}}{\partial \mathbf{v}_i} = \mathbf{v}_u. \tag{3}$$

We can see that, for the different preferences of $i_a$ and $i_b$ of a user $u$, the gradient in the loss function is the same $\mathbf{v}_u$, which is used to correct the prediction bias. In other words, our confidence about the preference for every item is the same, which is clearly not the case in practice. We believe that an item with higher preference reveals more reliably the true interest of a user, and it should cost more if the item is predicted wrong. This stronger feedback about the item should have a higher impact on the model update. The GPR-OPT criterion, which is described in the next section, aims to produce this effect.

## 3. GPR-OPT

We assume the preference of a user toward items follows the Gaussian distribution. Let us provide a brief introduction to the Gaussian model, before we introduce our proposed Gaussian optimization criterion GPR-OPT.

### 3.1. Gaussian distribution

In probability theory, a Gaussian distribution is a type of continuous probability distribution for a real-valued random variable (Ross, 2014). The general form of its probability density function is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2), \tag{4}$$

where $x$ is the random variable, the parameter $\mu$ is the mean or expectation of the distribution, $\sigma$ is its standard deviation.

### 3.2. Multivariate Gaussian distribution

For multivariate cases, we assume each dimension is independent, we can extend the variable $x$ in the above formula into multivariate $\mathbf{x} = \{x_1, x_2, \dots, x_k\}$:

$$f_{\mathbf{X}}\left(x_1, \dots, x_k\right) = \prod_{i=1}^{k} \frac{1}{\sigma_i \sqrt{2\pi}} \exp(-\frac{1}{2}\left(\frac{x_i - \mu}{\sigma_i}\right)^2), \tag{5}$$

where $x_i$ is one dimension of multivariate $\mathbf{x}$.

### 3.3. The Gaussian optimization criterion

Considering the preference of a user toward items is influenced by many unknown random variables, we assume it follows multivariate Gaussian distribution, which has been proved widely exists in most cases according to Central Limit Theorem (Rosenblatt, 1956).

Given a user $u$ and item $i$, represented as $\mathbf{v}_u \in R^h$ and $\mathbf{v}_i \in R^h$ respectively, we assume that each dimension in $\mathbf{v}_u$ is independent for simplicity. We can generate the preference distribution of items from user features, and the parameters of the multivariate Gaussian distribution is calculated by:

$$\mathbf{v}_i \sim \mathcal{N}(\boldsymbol{\mu}, diag(\boldsymbol{\sigma}^2)), \tag{6}$$

$$\boldsymbol{\mu} = \mathbf{W}_{\mu}^{\top} \cdot \mathbf{v}_u, \tag{7}$$

$$\log \boldsymbol{\sigma}^{\mathbf{2}} = \mathbf{W}_{\sigma}^{\top \cdot} \mathbf{v}_u, \tag{8}$$

where the $\mu$ and $\log \sigma^2$ are generated from user feature,[4] and the item feature $\mathbf{v}_i$ follows the multivariate Gaussian distribution derived from $\mathbf{v}_u$.

Based on the above settings, we can use the probability density of multivariate Gaussian distribution as the final prediction result. The predicted probability of user $u$ toward item $i$ can be formulated as:

$$\hat{p}_g(u, i) = \prod_{j=1}^{h} \frac{1}{\sigma_j \sqrt{2\pi}} \exp(-\frac{(v_{ij} - \mu_j)^2}{2\sigma_j^2}), \tag{9}$$

where $v_{ij}$ is the value of $j$th dimension in the item representation $\mathbf{v}_i$, $\mu_j$ and $\sigma_j$ are the mean and standard deviation of Gaussian distribution generated from the $j$th dimension of user $u$'s representation $\mathbf{v}_u$, and $h$ is the dimension size of the representations of users and items.

Following the best practice (Kingma & Welling, 2014; Mnih & Salakhutdinov, 2008), we maximize the log-likelihood function of the probability $\hat{p}_g$ by:

$$\begin{aligned} \hat{p}_{GPR}(u, i) &= \log \hat{p}_g(u, i) \\ &= \log(\prod_{j=1}^{h} \frac{1}{\sigma_j \sqrt{2\pi}} \exp(-\frac{(v_{ij} - \mu_j)^2}{2\sigma_j^2})) \\ &= -\frac{1}{2} \sum_{j=1}^{h} (\frac{(v_{ij} - \mu_j)^2}{\sigma_j^2} + \log \sigma_j^2 + \log 2\pi), \end{aligned} \tag{10}$$

From Eq. (10), we can see that compared with the dot product similarity function, GPR-OPT can distinctively learn the different user preferences towards different items, and will return larger gradients to punish the prediction bias when the user has higher confidence in the item (smaller $\sigma$). We will give detailed demonstrations and explanations in the following section.

## 4. The validation of GPR-OPT

In this section, we will show how GPR-OPT learns the explicit confidence of users' preferences from implicit feedback and solves the indiscriminate problem in Generic RS.

[4] Following Kingma and Welling (2014), we calculate $\log \sigma^2$ instead of $\boldsymbol{\sigma}^2$, because $\boldsymbol{\sigma}^2$ is always non-negative and cannot be learned directly, while $\log \boldsymbol{\sigma}^2$ does not have this restriction.

### 4.1. Preference confidence learning

The parameters $\mu$ and $\sigma$ in the Gaussian distribution of user preference are generated from the original representation of user $u$. They can be used as follows:

- The parameter $\mu$ can be used to represent the transferred user representation (see in Eq. (7)). As the training process, the representation of the positive item $\mathbf{v}_i$ will be approaching the transferred user representation $\mu$ to maximize the probability function (see in Eq. (10));
- The parameter $\sigma$ can be used to represent the preference confidence of users toward the item because it is quite reasonable that for the items with the same distance to $\mu$, the higher confidence (*i.e.,* smaller $\sigma$) in Gaussian distribution, the larger punish gradient to reduce the prediction bias.

We further validate the learning of the user's preference confidence by calculating the gradients of GPR-OPT. Given an interaction pair $u$ and $i$ (represented as $\mathbf{v}_u$ and $\mathbf{v}_i$), for dimension $j$ of the transferred user vector $\boldsymbol{\mu}$ from $\mathbf{v}_u$, we have:

$$\frac{\partial \hat{p}_{GPR}}{\partial \mu_j} = \frac{v_{ij} - \mu_j}{\sigma_j^2}. \tag{11}$$

**Discussion** : Different from the generic RS framework in implicit recommender systems (see in Eq. (3)), in which the user's confidence about the preference for different item $i_a$ and $i_b$ is the same. In the above Eq. (11), we can see that by maximizing $\hat{p}_{GPR}$, the item representation $v_{ij}$ and transferred user representation $\mu_j$ will be updated to become closer. The step we take to update $v_{ij}$ and $\mu_j$ is influenced by $\sigma_j$. If the distance between $v_{ij}$ and $\mu_j$ is large, then $\sigma_j$ will become larger to alleviate the punishing gradient, which enables GPR-OPT to treat items of different importance differently, and meanwhile learn the explicit user preference confidence from implicit feedback. The higher the preference confidence, the smaller the value of $\sigma$, and the GPR-OPT will return larger gradients to punish the prediction bias.

### 4.2. Indiscriminate problem in generic RS

In GPR-OPT, our aim is to optimize $\hat{p}_{GPR}$ in Eq. (10). Given a user $u$ and an item $i$, for a dimension $j$ in $\mathbf{v}_i$, we have:

$$\frac{\partial \hat{p}_{GPR}}{\partial v_{ij}} = \frac{\mu_j - v_{ij}}{\sigma_j^2} \tag{12}$$

**Discussion** : we can see that for the items $i_a$ and $i_b$ with the different preferences of a user $u$, the gradients in GPR-OPT are different, which helps to avoid the indiscriminate problem of items in the generic RS framework (see in Eq. (2)). Furthermore, by modeling the preference confidence $\sigma$, GPR-OPT has the ability to assign more attention to the important items, *i.e.,* items with high preference confidence.

## 5. Experiments

We evaluate the performance of GPR-OPT on three real-world datasets and use it to optimize the typical existing CF algorithms, showing the superiority and universality of our proposed optimization criterion for implicit CF recommender systems.

### 5.1. Experimental settings

**Datasets.** We experiment with three representative real-world datasets Movielens, Amazon Book, and Yelp. Table 1 summarizes the statistics of the datasets.

- **MovieLens 1M**[5]: A widely used benchmark dataset for evaluating collaborative filtering algorithms. Each user has at least 20 ratings.
- **Amazon Book**[6]: Amazon-review is a widely used dataset for product recommendation. We select Amazon-book from the collection. Since it is unreliable to include users with few purchase times, we use the 15-core setting, i.e., retaining users and items with at least 15 interactions.
- **Yelp 2018**[7]: contains businesses, reviews, and user data information. We use the 15-core setting, i.e., retaining users and items with at least 15 interactions.

**Compared Methods**. To intuitively demonstrate the effectiveness of confidence learning in GPR-OPT, we make comparisons with the matrix factorization methods with different confidence learning strategies, including:

5 https://grouplens.org/datasets/movielens/1m/.
6 http://deepyeti.ucsd.edu/jianmo/amazon/index.html.
7 https://www.yelp.com/dataset/challenge.

**Table 1**
Statistics of the datasets.

| Datasets | #Users | #Items | #Interactions | Density |
|---|---|---|---|---|
| MovieLens 1M | 6040 | 3706 | 1,000,209 | 4.468% |
| Amazon Book | 68,497 | 126,556 | 2,954,716 | 0.034% |
| Yelp 2018 | 60,642 | 88,097 | 2,284,811 | 0.042% |

- **WMF** (Hu et al., 2008): uses a constant parameter to control the different confidence levels of users' preference according to the frequency of the interactions with items.
- **BPR** (Rendle et al., 2009): is a classic matrix factorization collaborative filtering model and it is optimized with a pairwise ranking loss.
- **GER** (Dos Santos et al., 2017): assumes the vectors of users and items follow Gaussian distribution respectively, and further uses the variance in user and item vectors to compute the preference uncertainty.
- **GPR-OPT (ours)**: replaces the prediction by dot product operation of user and item vectors in MF with Gaussian density-based learning function.

To make fair comparisons, for the complex neural CF models (Chen et al., 2020; He et al., 2017; Wang et al., 2019) whose performance is also influenced by the deep neural networks in model architecture, we adopt GPR-OPT to replace the final interaction function (*i.e.,* MLP interaction function in NCF, and dot product function in graph-based CF models). The methods compared include:

- **NCF v.s. NCF$_{\mathbf{GPR-OPT}}$**. NCF (He et al., 2017) is a typical neural MF-based RS model, which is composed of deep and general MF. NCF$_{\text{GPR-OPT}}$ works on the general MF part by replacing the dot product with the Gaussian density function.
- **NGCF v.s. NGCF$_{\mathbf{GPR-OPT}}$**. NGCF (Wang et al., 2019) is a representative neural graph-based collaborative filtering model, and it utilizes a specialized graph convolution operation to capture the neighborhood collaborative filtering signal. NGCF$_{\text{GPR-OPT}}$ feeds the embeddings of the user and item into the GPR component to obtain the interaction probability.
- **LightGCN v.s. LightGCN$_{\mathbf{GPR-OPT}}$**. LightGCN (He et al., 2020) is the most recent graph-based CF RS model. It learns user and item embeddings by linearly propagating them on the user–item interaction graph and uses the weighted sum of the embeddings learned at all layers as the final embedding. LightGCN$_{\text{GPR-OPT}}$ stack GPR component to learn from the final embeddings.

By replacing the element-wise product operation in the above approaches with the Gaussian density function (see in Eq. (10)), our proposed GPR-OPT becomes a Gaussian matrix factorization model. GPR-OPT can also be used as a basic component to model the interaction between the user and the item, making it possible to assign a high learning weight to the items the user truly likes.

**Evaluation Metrics.** To avoid sample bias in the evaluation of RS (Krichene & Rendle, 2020), we evaluate our model on all candidate items. Given a user and a positive item, we use all items that the user had not interacted with as negative items.[8] For recommendation, the system will produce a ranking list of all items for a user. We apply two widely used metrics to evaluate the performance: Hit ratio at rank $k$ (Hit@$k$) and Normalized Discounted Cumulative Gain at rank $k$ (NDCG@$k$). For each method, a ranking list of all evaluated items is generated for evaluation.

**Parameter Settings**. For each dataset, we split the historical interactions of each user in chronological order into 8:1:1 for training, validating, and testing. For each positive item, we sample 5 items that the user had not interacted with as negative instances. We report the result of each method with its optimal hyperparameter settings on the validation data. For each baseline method, a grid search is applied to find the optimal settings. These include latent dimensions $h$ from $\{32, 64, 128, 256\}$, and the learning rate from $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$. For fair comparisons, the parameters in all the compared methods optimized by GPR-OPT are the same as the optimal baseline methods. The hyperparameters of each method are as follows: (1) BPR: the latent dimensions are 128, 64, and 64 in MovieLens 1M, Amazon Book, and Yelp, and the learning rates are 0.001, 0.01, and 0.001 respectively. (2) NCF: the embedding sizes are 128 and 64, the learning rate is 0.001 in all datasets, the number of hidden layers in MLP is three, and the dimension in each MLP layer is set to half of the previous layer. (3) NGCF: the embedding size of nodes is 8, and the learning rates are 0.001 in all datasets. (4) LightGCN: the embedding size is 128, the learning rate is 0.001, and the number of layers is 2 in all datasets.

### 5.2. Main results

We present the results of Hit@$k$ and NDCG@$k$ ($k$ = 20 and $k$ = 50) on the test set in Table 2. We focus on analyzing the advantages of GPR-OPT to learn different performance confidences of items, as well as the effectiveness of the utilization as the optimization criterion on existing CF methods (see Tables 3–5). We have the following observations:

(1) Our proposed Gaussian matrix factorization model GPR-OPT performs the best on three datasets due to its ability to learn the explicit confidence of interacted items. The slight improvements on the Yelp dataset may be caused by that in addition to the

[8] For the graph-based model NGCF, we evaluate it with 1000 negative items due to the large memory occupation. Others conduct evaluations on all candidate items.

**Table 2**
Performance comparison of different methods with preference confidence learning.

| Datasets | Models | WMF | BPR | GER | GPR-OPT | Gain |
|---|---|---|---|---|---|---|
| MovieLens 1M | Hit@20 | 0.1012 | 0.1769 | 0.0665 | **0.2004** | **+10.04%** |
| | Hit@50 | 0.1874 | 0.2532 | 0.1261 | **0.2681** | **+3.880%** |
| | NDCG@20 | 0.0432 | 0.0994 | 0.0266 | **0.1206** | **+22.69%** |
| | NDCG@50 | 0.0601 | 0.1144 | 0.0384 | **0.1339** | **+17.05%** |
| Amazon Book | Hit@20 | 0.0105 | 0.0126 | 0.0040 | **0.0133** | **+5.56%** |
| | Hit@50 | 0.0237 | 0.0257 | 0.0084 | **0.0270** | **+5.06%** |
| | NDCG@20 | 0.0039 | 0.0048 | 0.0015 | **0.0050** | **+4.16%** |
| | NDCG@50 | 0.0065 | 0.0073 | 0.0025 | **0.0077** | **+5.48%** |
| Yelp 2018 | Hit@20 | 0.0076 | 0.0238 | 0.0092 | **0.0242** | **+1.68%** |
| | Hit@50 | 0.0196 | 0.0495 | 0.0179 | **0.0509** | **+2.83%** |
| | NDCG@20 | 0.0026 | 0.0090 | 0.0036 | **0.0090** | **+0.00%** |
| | NDCG@50 | 0.0049 | 0.0140 | 0.0053 | **0.0142** | **+1.42%** |

**Table 3**
Performance comparison of different methods optimized by GPR-OPT on MovieLens 1M dataset.

| MovieLens 1M | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | NCF | NCF$_{GPR}$ | Gain | NGCF | NGCF$_{GPR}$ | Gain | LightGCN | LightGCN$_{GPR}$ | Gain |
| Hit@20 | 0.2171 | 0.2389 | **+10.04%** | 0.1876 | 0.1908 | **+1.706%** | 0.1837 | 0.1906 | **+3.76%** |
| Hit@50 | 0.2787 | 0.2895 | **+3.88%** | 0.3459 | 0.3512 | **+1.532%** | 0.2568 | 0.2673 | **+4.09%** |
| NDCG@20 | 0.1441 | 0.1768 | **+22.69%** | 0.0777 | 0.0790 | **+1.673%** | 0.1066 | 0.1105 | **+3.66%** |
| NDCG@50 | 0.1563 | 0.1868 | **+17.05%** | 0.1089 | 0.1106 | **+1.561%** | 0.1210 | 0.1256 | **+3.80%** |

**Table 4**
Performance comparison of different methods optimized by GPR-OPT on Amazon Book dataset.

| Amazon Book | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | NCF | NCF$_{GPR}$ | Gain | NGCF | NGCF$_{GPR}$ | Gain | LightGCN | LightGCN$_{GPR}$ | Gain |
| Hit@20 | 0.0113 | 0.0135 | **+19.47%** | 0.2770 | 0.2888 | **+4.260%** | 0.0165 | 0.0193 | **+16.97%** |
| Hit@50 | 0.0237 | 0.0273 | **+15.19%** | 0.4401 | 0.4538 | **+3.043%** | 0.0322 | 0.0380 | **+18.01%** |
| NDCG@20 | 0.0042 | 0.0052 | **+23.81%** | 0.1207 | 0.1266 | **+4.890%** | 0.0044 | 0.0074 | **+15.63%** |
| NDCG@50 | 0.0066 | 0.0079 | **+19.70%** | 0.1530 | 0.1592 | **+4.052%** | 0.0095 | 0.0111 | **+16.84%** |

**Table 5**
Performance comparison of different methods optimized by GPR-OPT on Yelp 2018 dataset.

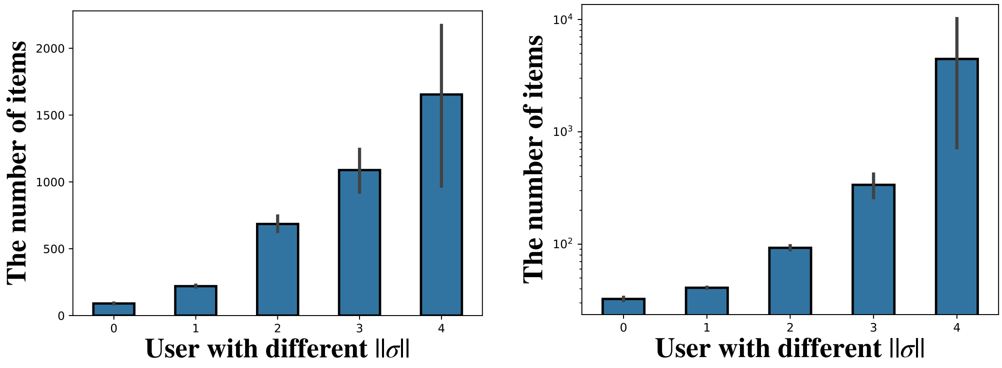
| Yelp 2018 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | NCF | NCF$_{GPR}$ | Gain | NGCF | NGCF$_{GPR}$ | Gain | LightGCN | LightGCN$_{GPR}$ | Gain |
| Hit@20 | 0.0218 | 0.0275 | **+26.15%** | 0.4756 | 0.5043 | **+6.034%** | 0.0282 | 0.0326 | **+15.60%** |
| Hit@50 | 0.0467 | 0.0538 | **+15.20%** | 0.6713 | 0.6999 | **+4.260%** | 0.0109 | 0.0125 | **+14.67%** |
| NDCG@20 | 0.0082 | 0.0108 | **+31.71%** | 0.2265 | 0.2392 | **+5.607%** | 0.0562 | 0.0661 | **+17.62%** |
| NDCG@50 | 0.0130 | 0.0160 | **+23.08%** | 0.2654 | 0.2781 | **+4.785%** | 0.0164 | 0.0192 | **+17.07%** |

preference of the user himself, the preference of the user is largely influenced by other social users in the Yelp dataset, in which the user shares reviews about restaurants, hotels, and so on.

(2) Compared with BPR, the performance of another Gaussian embedding model GER is very poor on three datasets, especially for the datasets with high sparsity (Amazon and Yelp), indicating that the uncertainty modeled in the representation of user/item vectors independently cannot effectively be captured in learning the uncertainty of user preference, on the contrary, will increase the difficulty of training by incorporating more variables. Our GPR-OPT uses Gaussian distribution to directly model the performance uncertainty (confidence), which can be effectively learned to improve the model performance.

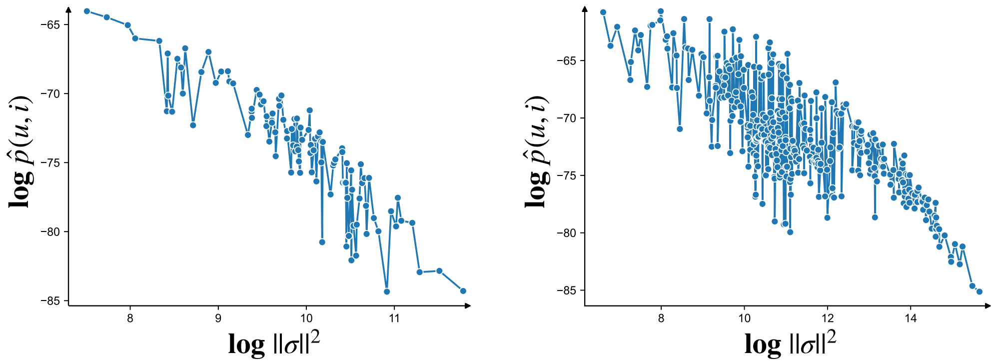
(3) For the weighted matrix factorization model WMF, we find that assigning different confidence levels of items by constant weights does not work well for recommender systems. It is impossible to make the constant weight decided by the frequency of items fit all the users, which results in the poor performance of WMF.

(4) Our GPR-OPT is a light optimization criterion and can be easily adopted into existing collaborative filtering models, *e.g.,* NCF, NGCF, and LightGCN. Optimized by GPR-OPT, almost all the baseline methods gain significant improvements (with average 19.00%, 3.62%, and 12.31% improvements of NCF, NGCF, and LightGCN respectively on all datasets), showing the effectiveness of learning the confidence of user preference in implicit RS. We find that the improvements of GPR-OPT are less in the NGCF model than in other methods, especially on the MovieLens 1M dataset, this may be caused by that the collaborative filtering signal had been well captured in the NGCF model, but on datasets with higher data sparsity, the ability of NGCF decreases, making the improvement spaces for our optimization criterion GPR-OPT.




(a) MovieLens 1M (b) Amazon Book

**Fig. 2.** Identifying Gaussian distribution in GPR-OPT with the statistics of user-adopted items.




(a) MovieLens 1M (b) Amazon Book

**Fig. 3.** The relationship between confidence and preference of users.

(5) The average improvement of models with GPR-OPT is 13.50% and 15.51% in Amazon Book and Yelp datasets, exceeding the 6.32% in Movielens 1M dataset. This implies that our GPR-OPT can be well adopted in large-scale recommendation scenarios with high data sparsity.

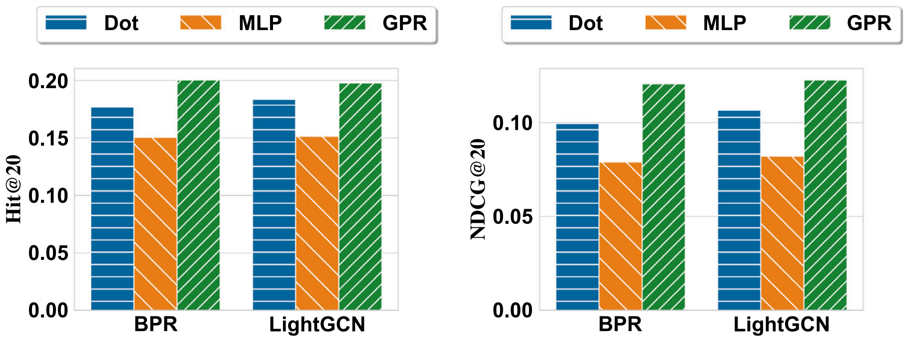
### *5.3. Experimental analysis*

In this section, we analyze the Gaussian distribution of user preference in GPR-OPT and compare GPR-OPT with the complex interaction function, like Multi-Layer Perception (MLP).

**Interpretability of Gaussian Distribution in GPR-OPT**. In GPR-OPT, the preference of a user toward items follows a multivariate Gaussian distribution. $\boldsymbol{\sigma}$ is the standard deviation vector of the preference distribution. The smaller $\|\boldsymbol{\sigma}\|$ is, the more centralized the distribution. We use the number of different items that the user had adopted to evaluate his preference distribution. A large number of different items the user liked means a wide range of his interests toward items. We divide the users into five bins with increasing values of $\|\boldsymbol{\sigma}\|$, labeled from $[1, 5]$. As shown in Fig. 2, we can see that: the more the user adopted different items, the wider the distribution of user preference, *i.e.,* larger $\|\boldsymbol{\sigma}\|$. The results are consistent between the two datasets, which demonstrates the reasonableness of the multivariate Gaussian distribution learned in GPR-OPT.

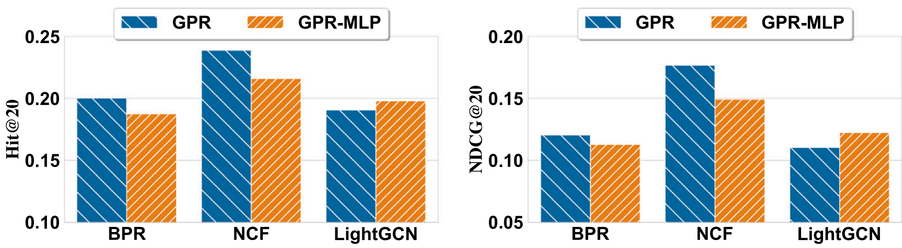
**The Relationship between Confidence and Preference in GPR-OPT**. In GPR-OPT, $\boldsymbol{\sigma}$ represents the standard deviation vector of the preference distribution. The smaller $\|\boldsymbol{\sigma}\|$ is, the more confidence in the user preference. To keep the consistency with the value in the log-likelihood optimization function, we show the relationship between confidence parameter $\log \|\boldsymbol{\sigma}\|^2$ and the user preference $\log \hat{p}(u, i)$ in Fig. 3, we can see that the prediction of user preference $\hat{p}(u, i)$ increases as the confidence becomes larger in the MovieLens and Amazon datasets, which demonstrates the effectiveness of proposed GPR-OPT in capturing user confidence in implicit recommender systems.

**Comparing with MLP Interaction Function.** In our experiments, we compare GPR-OPT with the most widely used interaction function in generic RS, *i.e.,* simple dot product. We also conduct experiments on baseline methods with a complex interaction




(a) Hit@20 (b) NDCG@20

**Fig. 4.** Identifying Gaussian distribution in GPR-OPT with the statistics of user-adopted items.




(a) Hit@20 (b) NDCG@20

**Fig. 5.** The comparison between models optimized by GPR-OPT and its variant GPR-OPT-MLP.

function, *e.g.,* MLP. Given the user and item representation $\mathbf{v}_u$ and $\mathbf{v}_i$, the final prediction probability obtained by MLP interaction function by:

$$\hat{p}_{MLP}(u,i) = \phi_{MLP_{(n)}}(\mathbf{v}_u \oplus \mathbf{v}_i), \tag{13}$$

where $\phi$ is the activation function ReLU, $\oplus$ is the concat operation of two vectors, $n$ is the layer number of MLP. We set $n = 3$ in our experiments to obtain optimal results. As shown in Fig. 4, we can see that replacing the simple dot product function (see in Eq. (2)) with the complex MLP function (see in Eq. (13)) leads decreased performances on BPR and LightGCN. Gaussian-based interaction function GPR-OPT achieves the best performance compared with both simple dot product and complex MLP interaction functions.

**Optimization of GPR-OPT.** In GPR-OPT, we assume that each dimension in user representation is independent, and calculate the final probability by simple summation (see Eq. (2)). Considering the assumption may limit the capability of our model, we test another complex interaction function *e.g.,* MLP, on the vector of the probability density of all dimensions to capture the relations between them. We set the number of MLP layers to 3 in our experiments to obtain the optimal results. The results are shown in Fig. 5, we can see that the results on different models are not consistent: for BPR and NCF, the complex interactions by MLP decrease the performance of the GPR-OPT model, while for LightGCN, the performance slightly improves over the model with GPR-OPT. This result shows that the independence assumption between dimensions may be reasonable in practice, and a method trying to make the dimensions dependent is not always better. More investigations are needed to better understand the relations between dimensions in the future.

## 6. Related work

Recommender systems have attracted a lot of attention from the research community and industry. We first introduce the challenges in implicit recommender systems, and then briefly introduce the mainstream recommendation algorithms.

### *6.1. Implicit RS algorithms*

In recommender systems (RS), the preferences of users are inferred by the feedback of users, which could be either implicit or explicit (Chen & Peng, 2018; Hsu, Yeh, et al., 2018; Mandal & Maiti, 2020; Parra et al., 2011). Both explicit and implicit feedback can reflect users' tastes in items, which are essential for predicting the user preferences (Jia & Wang, 2022; Lian et al., 2018; Zhang et al., 2014). Both types of feedback have been leveraged in RS. In explicit feedback, the preference of the user can be explicitly expressed, including assigned ratings, tags, and other personalized information. For example, the user's explicit behavior information is utilized in Liao, Deng, Wan, and Liu (2022) and Xie et al. (2020) to enhance the prediction of user preference. Ranking and rating prediction tasks are jointly optimized in Jadidinejad et al. (2019) by learning from both explicit rating and implicit click feedback. In the most real-world scenarios, the explicit preference of users is always incomplete due to the individual privacy issue and the high cost of human resources. In contrast, implicit feedback, such as a click or viewing records of users, is more frequently available for recommender systems. However, in implicit RS, the unknowable attitude of a user toward the adopted items makes it difficult to predict the preference of users (Parra et al., 2011; Wang et al., 2023). Most existing studies in implicit feedback RS (He et al., 2020, 2017; Kang & McAuley, 2018; Rendle et al., 2009) treat all interacted items as his favorite taste and assign a uniform confidence weight for all positive user–item interactions. One typical implicit recommendation algorithm is MF-based collaborative filtering methods (He et al., 2017; Kutlimuratov, Abdusalomov, Oteniyazov, Mirzakhalilov, & Whangbo, 2022; Mnih & Salakhutdinov, 2008; Rendle et al., 2009). They model the interaction between the latent vectors of the user and the item. Another is the graph-based RS algorithms (Chen et al., 2021; Dridi, Tamine, & Slimani, 2022; Hamilton, Ying, & Leskovec, 2017; He et al., 2020; Li, Ren, & Deng, 2022; Liu, Cheng, Zhu, Gao, & Nie, 2021; Wang et al., 2019; Ying et al., 2018), which learn the representation of user and item on a unified graph by aggregating the collaborative high-order neighborhood information from the graph by convolutional networks. Besides, some efforts have been made to make utilization of the negative feedback in implicit recommendations, for example, Li et al. (2022) address the negative preferences in implicit feedback data via generative adversarial networks to solve the fairness issue in recommender systems.

### 6.2. Learning preference confidence

Some efforts had been made in previous studies to incorporate different preference confidence of users towards different items. Some studies (Coscrato & Bridge, 2023; Hu et al., 2008; Liang et al., 2016; Yang et al., 2022) use a constant parameter to control the confidence levels of users' preference according to the frequency of the interactions with items. Probability-based methods (Dos Santos et al., 2017; Jiang et al., 2019; Li, Chin, Chen, & Cong, 2021) generate data from a suitable probability distribution, which is parameterized by some low-dimensional latent factors. Among them, the data confidence is also addressed in some literature (Chen et al., 2019; Wang et al., 2023) to optimize the modeling process in recommender systems. Besides, Gaussian embedding methods (Dos Santos et al., 2017; Jiang et al., 2019), in which the representations of users/items are flexible with learned uncertainty variables, can also learn the preference confidence from the uncertainty of user/item representations. Instead of using the Gaussian distribution as the data distribution, the multinomial distribution has been used for discrete data (Deng, Jing, Yu, Sun, & Zhou, 2018; Liang, Krishnan, Hoffman, & Jebara, 2018; Sedhain, Menon, Sanner, & Xie, 2015; Wang, 2021; Zhao et al., 2020) for recommendation. Besides, adopting the neural interaction networks, *e.g.,* MLP-based learned function, can also automatically learn the preference confidence from input features (Chen et al., 2020; He et al., 2017).

Distinguishing preference confidence of users can be learned in these models, but still in an inexplicable way, that is to say, they cannot explicitly learn the preference of users in implicit feedback recommender systems, leading to poor interpretability of the model. Some previous work (Parra et al., 2011) maps implicit feedback to explicit one via a logistic regression model. However, the model still needs to be supervised by explicit feedback labels. Few previous studies have addressed this problem without any extra information.

### 6.3. Gaussian representation learning

As the Gaussian embeddings are successfully employed to learn the representations of word (Vilnis & McCallum, 2014), it has been also utilized in recommender systems (Dos Santos et al., 2017; Fan, Liu, Wang, Zheng, & Yu, 2021; Hoang, Deoras, Zhao, Li, & Karypis, 2022; Jiang et al., 2019; Yang, Liu, & Liu, 2021). The variance of the Gaussian distribution is used to measure the uncertainty of the user/item representations, and the score of the label is estimated along with their confidence. For example, Yang et al. proposed a variational autoencoder that uses a Gaussian mixture model for latent factors distribution for collaborative filtering recommendations. They focus on learning better representation of users and items, and cannot directly learn the preference confidence about the adopted items. Fan et al. propose an elliptical Gaussian distribution-based method for sequential recommendation and they inject the uncertainties into sequential modeling.

Different from the above Gaussian embedding-based RS, in our paper, we use Gaussian distribution to directly model the preference confidence of users toward items. This makes the preference confidence of the user learnable in implicit RS. This also makes it easy to adopt GPR-OPT into most existing recommendation models. So the GPR-OPT criterion is complementary to many existing approaches to RS and can be used to enhance the latter, as we showed in this paper.

## 7. Conclusion

This paper highlights the critical issue in recommender systems with implicit feedback. We propose a practical Gaussian-based optimization criterion GPR-OPT for implicit recommender systems. we describe users in high-dimensional continuous space and aim to learn a preference distribution of items in the space of users. Our optimization converts the implicit feedback to learnable explicit feedback with self-adaptive confidence and can be easily adapted into most existing recommendation models. We show in our experiments that GPR-OPT can generally positively impact the existing recommender systems. While the idea of optimizing the existing algorithms with only the ID feature of users and items is validated, our implementation can be further improved. For example, we can detect the optimization of RS models with more features of users and items. Besides, although mapping user preference of items with Gaussian distribution can be adopted in most of the cases in real-world recommendation scenarios, in the cases with prior knowledge of data distribution, we will make an attempt to design an automatic way to learn the specific optimization function with the prior knowledge of the different data distribution.

## CRediT authorship contribution statement

**Ting Bai:** Conceptualization, Methodology, Writing. **Xiaotong Wang:** Methodology, Validation. **Zhenhao Zhang:** Methodology, Formal analysis. **Wei Song:** Data curation. **Bin Wu:** Supervision. **Jian-Yun Nie:** Writing – review & editing.

## Data availability

Data will be made available on request.

## References


Chen, C., Ma, W., Zhang, M., Wang, Z., He, X., Wang, C., et al. (2021). Graph heterogeneous multi-relational recommendation. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35* (pp. 3958–3966).

Chen, S., & Peng, Y. (2018). Matrix factorization for recommendation with explicit and implicit feedback. *Knowledge-Based Systems*, *158*, 109–117.

Chen, J., Wang, C., Zhou, S., Shi, Q., Chen, J., Feng, Y., et al. (2020). Fast adaptively weighted matrix factorization for recommendation with implicit feedback. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34* (pp. 3470–3477).

Chen, J., Wang, C., Zhou, S., Shi, Q., Feng, Y., & Chen, C. (2019). Samwalker: Social recommendation with informative sampling strategy. In *The world wide web conference* (pp. 228–239).

Coscrato, V., & Bridge, D. (2023). Recommendation uncertainty in implicit feedback recommender systems. In *Artificial intelligence and cognitive science: 30th irish conference, AICS 2022, Munster, Ireland, December 8–9, 2022, revised selected papers* (pp. 279–291). Springer.

Deng, D., Jing, L., Yu, J., Sun, S., & Zhou, H. (2018). Neural gaussian mixture model for review-based rating prediction. In *Proceedings of the 12th ACM conference on recommender systems* (pp. 113–121).

Dos Santos, L., Piwowarski, B., & Gallinari, P. (2017). Gaussian embeddings for collaborative filtering. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 1065–1068).

Dridi, R., Tamine, L., & Slimani, Y. (2022). Exploiting context-awareness and multi-criteria decision making to improve items recommendation using a tripartite graph-based model. *Information Processing & Management*, *59*(2), Article 102861.

Fan, Z., Liu, Z., Wang, S., Zheng, L., & Yu, P. S. (2021). Modeling sequences as distributions with uncertainty for sequential recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 3019–3023).

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in neural information processing systems* (pp. 1024–1034).

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 639–648).

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173–182).

Hoang, N., Deoras, A., Zhao, T., Li, J., & Karypis, G. (2022). Learning personalized item-to-item recommendation metric via implicit feedback. In *International conference on artificial intelligence and statistics* (pp. 1062–1077). PMLR.

Hsu, C.-C., Yeh, M.-Y., et al. (2018). A general framework for implicit and explicit social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *30*(12), 2228–2241.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 eighth IEEE international conference on data mining* (pp. 263–272). IEEE.

Jadidinejad, A. H., Macdonald, C., & Ounis, I. (2019). Unifying explicit and implicit feedback for rating prediction and ranking recommendation tasks. In *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval* (pp. 149–156).

Jia, Y., & Wang, H. (2022). Learning neural ranking models online from implicit user feedback. In *Proceedings of the ACM web conference 2022* (pp. 431–441).

Jiang, J., Yang, D., Xiao, Y., & Shen, C. (2019). Convolutional Gaussian embeddings for personalized recommendation with uncertainty. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 2642–2648). AAAI Press.

Kang, W.-C., & McAuley, J. (2018). Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)* (pp. 197–206). IEEE.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. *Statistics*, *1050*, 10.

Krichene, W., & Rendle, S. (2020). On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1748–1757).

Kutlimuratov, A., Abdusalomov, A. B., Oteniyazov, R., Mirzakhalilov, S., & Whangbo, T. K. (2022). Modeling and applying implicit dormant features for recommendation via clustering and deep factorization. *Sensors*, *22*(21), 8224.

Li, X., Chin, J. Y., Chen, Y., & Cong, G. (2021). Sinkhorn collaborative filtering. In *Proceedings of the web conference 2021* (pp. 582–592).

Li, J., Ren, Y., & Deng, K. (2022). FairGAN: GANs-based fairness-aware learning for recommendations with implicit feedback. In *Proceedings of the ACM web conference 2022* (pp. 297–307).

Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., & Sun, G. (2018). xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1754–1763).

Liang, D., Charlin, L., McInerney, J., & Blei, D. M. (2016). Modeling user exposure in recommendation. In *Proceedings of the 25th international conference on world wide web* (pp. 951–961).

Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018). Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference* (pp. 689–698).

Liao, G., Deng, X., Wan, C., & Liu, X. (2022). Group event recommendation based on graph multi-head attention network combining explicit and implicit information. *Information Processing & Management*, *59*(2), Article 102797.

Liu, F., Cheng, Z., Zhu, L., Gao, Z., & Nie, L. (2021). Interest-aware message-passing gcn for recommendation. In *Proceedings of the web conference 2021* (pp. 1296–1305).

Mandal, S., & Maiti, A. (2020). Explicit feedback meet with implicit feedback in GPMF: a generalized probabilistic matrix factorization model for recommendation. *Applied Intelligence*, *50*(6), 1955–1978.

Mnih, A., & Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In *Advances in neural information processing systems* (pp. 1257–1264).

Parra, D., Karatzoglou, A., Amatriain, X., & Yavuz, I. (2011). Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. In *Proceedings of the CARS-2011, Vol. 5*.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence* (pp. 452–461). AUAI Press.

Rendle, S., Krichene, W., Zhang, L., & Anderson, J. (2020). Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM conference on recommender systems* (pp. 240–248).
Rosenblatt, M. (1956). A central limit theorem and a strong mixing condition. *Proceedings of the National Academy of Sciences of the United States of America*, *42*(1), 43.
Ross, S. M. (2014). *Introduction to probability models*. Academic Press.
Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on world wide web* (pp. 111–112).
Vilnis, L., & McCallum, A. (2014). Word representations via Gaussian embedding. arXiv preprint arXiv:1412.6623.
Wang, W. (2021). Learning to recommend from sparse data via generative user feedback. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35* (pp. 4436–4444).
Wang, C., Feng, F., Zhang, Y., Wang, Q., Hu, X., & He, X. (2023). Rethinking missing data: Aleatoric uncertainty-aware recommendation. *IEEE Transactions on Big Data*.
Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 165–174).
Xie, R., Ling, C., Wang, Y., Wang, R., Xia, F., & Lin, L. (2020). Deep feedback network for recommendation. In *Proceedings of IJCAI-PRICAI*.
Yang, F., Liu, F., & Liu, S. (2021). Collaborative filtering based on a variational Gaussian mixture model. *Future Internet*, *13*(2), 37.
Yang, T., Luo, C., Lu, H., Gupta, P., Yin, B., & Ai, Q. (2022). Can clicks be both labels and features? Unbiased behavior feature collection and uncertainty-aware learning to rank. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 6–17).
Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 974–983).
Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., & Ma, S. (2014). Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval* (pp. 83–92).
Zhao, H., Rai, P., Du, L., Buntine, W., Phung, D., & Zhou, M. (2020). Variational autoencoders for sparse and overdispersed discrete data. In *International conference on artificial intelligence and statistics* (pp. 1684–1694). PMLR.

**Ting Bai** received her Ph.D. degree from Renmin University of China in 2019. She currently is an assistant professor in the School of Computer Science, Beijing University of Posts and Telecommunications. Her major research interests are in recommender systems and Human behavior analysis. She has published several papers on SIGIR, WWW, KDD, CIKM, WSDM, TKDE, and so on. She is a member of the IEEE.