Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230929

The optimized probabilistic recommendation model in exposure bias

Yizhao Chen[1,5], Han Li[2], Ruiyang Liu[3] and Yuanbang Wu[4]

[1]Department of Computer Science, Northeastern University, Shenyang, 110819, China, 20206649@stu.neu.edu.cn
[2]Weitzman school of design, University of Pennsylvania, Philadelphia, 19104, America, seulihan@126.com
[3]WLSA Shanghai Academy, Shanghai,201900, China, elma20073411@gmail.com
[4]Department of Data Science, City University of Hong Kong, 999077, China, yuanbanwu2c@my.cityu.edu.hk

[5]20206649@stu.neu.edu.cn

Abstract: Recommendation models play a significant part on the internet. They combine user and information data to create personalized recommendations for various users. But this may be hidden by bias. Exposure bias is a kind of bias that blurs the distinction between a disliked item and an unshown one that may be likable. Countless research has been done to minimize this kind of bias. Our research will focus on using a new algorithm called" ExpoMF++", an enhanced version of Exposure Matrix Factorization. This enhancement is done in two ways: replacing the simple dot product with neural collaborative filtering and adding an optimization with the Gaussian mixture model. We tested this new model on different datasets of various sizes and attributes. Our model proves functional and works better than the original ExpoMF, significantly reducing problems caused by exposure bias. We then compared our model with other standard methods for exposure bias and found sound results.

Keywords: exposure bias, ExpoMF++, probabilistic recommendation model, Gaussian mixture model, Neural Collaborative Filtering.

1. Introduction

We live in the era of the Internet. When we open up an internet website, thousands of choices blossom in front of our eyes. But this surplus of information, or "Internet Overdose" creates a problem, as not all of it is what we want.

Websites, being the pesky businesses they are, will try to recommend the most relevant results to us. But even that is not easy, as recommendation results are often masked behind the bias. Bias limits what data recommendation systems see and use. There are many kinds of bias. Exposure bias is one of them. When a recommended item is not clicked, there is the possibility that the user receiving this recommendation does not like it and therefore did not choose it. But it may also be because the item is not shown to the user - it was not even" exposed". Since the first machine learning algorithms cannot tell it apart because both express an" unclicked" value, exposure bias hides the favorable results behind this" not exposed"-ness, skewing the recommendation results.

That is when recommendation models come in. By combining the user's personal information, the item's information, and the user's past interaction with the object through analysis using machine learning technology, they build a user interest model to provide users with precise recommendations and relieve people of the massive trouble of information selection. However, the total number of items and users in the recommendation process is too huge to be directly studied. To solve this, a smaller sample is obtained by sampling through the entire population, and the properties of the whole are estimated by generalization.

The chosen samples must represent the entire population to prevent large deviations between the estimated and actual values. [1] And besides, since models and evaluation metrics that purposefully ignore bias cannot reflect the performance of recommender systems because they become biased in the training process, it becomes crucial that recommendation systems be trained to address and debias the raw data before analyzing it. (Schnabel et al., Bonner and Vasile). This aspect, enhancing derbies learning, is the main direction of our research.

For directness, user feedback can be divided into explicit and implicit feedback.[2] We know that detailed data is accurate, providing a clear evaluation metric. But in implicit data, we can only know if the user clicked on the item, unable to determine whether the unclicked item is really an item the user does not like or just did not see. Therefore, two arisen questions must be answered: How to weigh the data with feedback? And what to do without it?

From our research, we discovered that most methods rely on giving more precise weighting, such as WMF weighing everything the same, complicated models such as CDAE using inspirative negative sampling to lower the influence of negative feedback involves live updating and cross-verification, unable to be realized in the billions of shifting data and weights in reality. To counter this, people invented Exposure Matrix Factorization - predicting user-item interactions to lessen negative feedback's consequences automatically. However, this method involves the evaluation of data exposure, and exposure bias still exists while updating.

We developed a probabilistic model named Exposure Matrix Factorization Plus (abbreviated as ExpoMF++) for recommendation on the basis of Exposure Matrix Factorization [3] (ExpoMF). ExpoMF++ is a probabilistic model combining heuristic and theoretical tools that capture whether an item is exposed to the user individually and whether the user decides to click on the item. This causes the algorithm to iterate between estimating user preference and estimating exposure, for example, why an item wasn't connected. When assessing preferences, it naturally lowers the weight of unclicked things users expect to like because it assumes that users will not be exposed to them.

Compared with the original model, the ExpoMF++ model has two aspects of optimization and reconstruction:

First, we use the Neural Collaborative Filtering framework model [4] to replace the original simple sigmoid function model that gets the result by dot product x of the exposure covariate. With deep refinements to model construction, we achieve a significant improvement in exposure priors as well as model performance.[5] We also conducted many real-life data experiments to compare and prove that the ExpoMF++ model has dramatically reduced exposure bias issues compared with the original model and other baseline methods.

The second is that the ExpoMF++ model reconstructs the algorithm of the original model and addsan optimization with the Gaussian mixture model [6-8]. The actual Bernoulli equation distribution and the assumption that $y_{ui}|a_{ui} = 1$ $N(\theta > u,\beta_i,\lambda{-}1y)$ significantly affect the model's performance. Without increasing the complexity of the algorithm, we replace the original with the Gaussian mixture model. After proving the algorithmic basis for the hypothesis, we then demonstrate the superiority of the new algorithmic model on semi-synthetic theoretical data and show its potential for improvement in the ExpoMF strategy.

In this research, we first review the background of Causal inference and the original ExpoMF. We then introduce the ExpoMF++ optimization in Section 3 and Hierarchical Modeling of Exposure in the subsequent subsection. We present the optimization of Learning Algorithms with Gaussian Mixed Model in Section 4. We show the related work, the connection between ExpoMF++, causal reasoning,

DOI: 10.54254/2755-2721/6/20230929

and other recommended research paths plotted in Section 5. Then we present an empirical study and the three research questions we made and solved. Finally, we summarize the conclusion in Section 7.

2. Background

As far away as recommendation systems first appeared, along came their long-term issue: bias. Exposure bias, one of four major types of discrimination, occurs when a user does not choose an item they favor simply because they were not exposed to it. This creates a huge problem, as unclicked items usually were assumed unfavorable. Just like recommendation systems, algorithms dedicated to the prevention of exposure bias have a long history. Previous works to optimize exposure bias mainly fall into the following four categories, and each of these methods devises different strategies to tackle the problem of exposure bias.

Heuristic-based methods[11-13] assume that user feedback depends on specific factors and try to simulate the effect of hidden factors; inverse propensity score-based methods use inverse propensity scores as sample weights to adjust for biased feedback distributions;unbiased data augmentation methods introduce a particular square target data to guide training with personal feedback assuming that the sample exposure or click obeys a Bernoulli distribution; and most importantly the theoretical tool-based approach aims to utilize combined theoretical tools to design debiased models, such as information bottlenecks and causal inference techniques[14,15].

We are diving deeper into the causal inference technique models in theoretical tool-based models.

3. Optimized exposure MF

We propose Exposure Matrix Factorization Plus (ExpoMF++). In Section 3.1, we explain and analyze our primary model. Section 3.2 discusses several ideas to incorporate optimization methods into ExpoMF++ and the optimized hierarchical modeling of exposures. Then, we discuss how to make predictions based on our model.

3.1. Model description

Only some items can be exposed, but not exposure does not represent negative preferences.

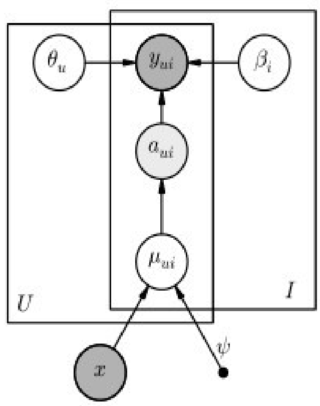

Figure 1. Before updating.

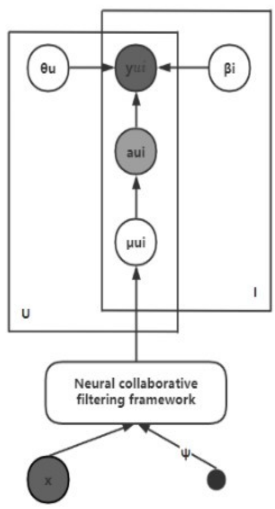

Figure 2. After updating.

The two figures are Graphical representations of the ExpoMF++ model (with the exposure covariate). Unshaded nodes indicate hidden variables. Shaded nodes represent observed variables. A directed edge from node A to node B indicates that variable B depends on variable A. A lightly shaded node a_{ui} indicates partial observance (i.e. it is observed when $y_{ui} = 1$, and not otherwise).

We use the Neural Collaborative Filtering framework model to replace the original simple sigmoid function model that obtains the result by dot product x of the exposure covariate. With deep refinements to model construction, we achieve a significant improvement in exposure priors as well as model performance.

For each combination of user $u = 1,...U$ and item $i = 1,...i$ .We consider two sets of variables. The first exposure matrix $A = \{a_{ui}\}$ represents whether user u has been exposed to item i, and the second interaction or "click" matrix $Y = \{y_{ui}\}$ represents whether user u has clicked on item i.Then We, have three refinements to the base model.

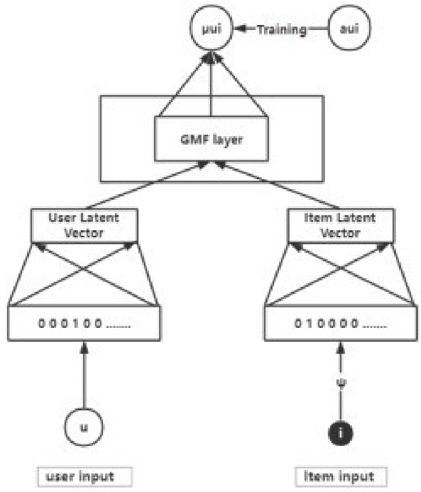

Figure 3. Adding GMF layer.

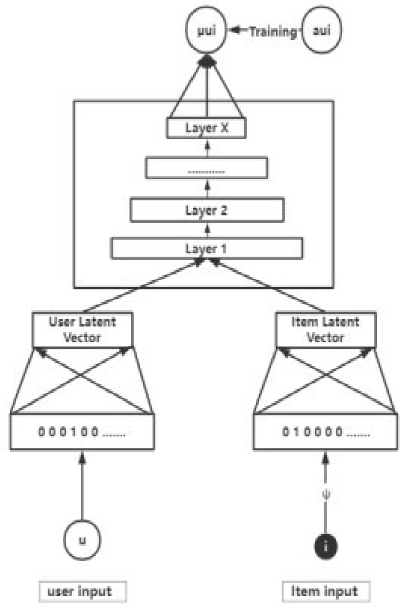

Figure 4. Adding MLP layer.

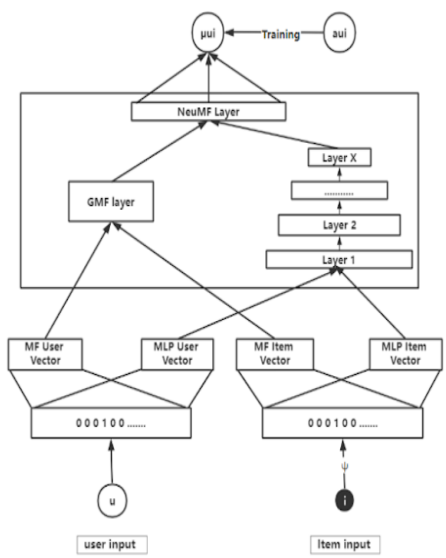

Figure 5. Adding NMF layer.

Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230929

3.2. Optimization hierarchical modeling of exposure

Now we discuss how to choose and learn *ui* after applying ExpoMF++. One straightforward way to encode exposure is through item popularity: not using any external information. We can fix the UI for all users and items at some global value, which means that only user factors, item factors, and clicks will determine exposure (conditioned on the variance hyperparameter).

The second way is through exposure covariates. Additional specific information containing the exposure rate, the exposure covariate, can help us fix a value μ_{ui} for particular values of u and i .

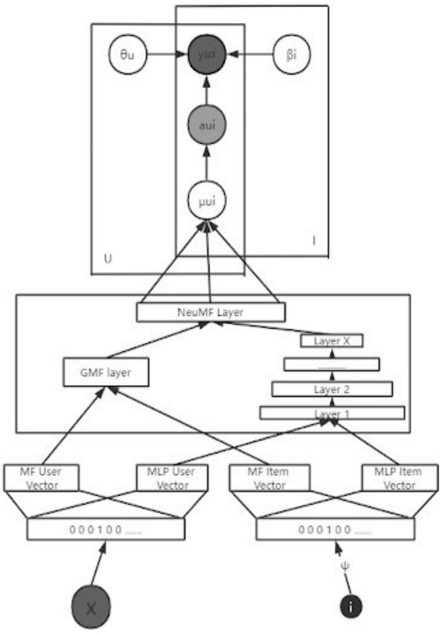

Figure 6. Combined Model.

In recommending text documents, we treat exposure covariates as the set of words for each record. In a location-based recommendation field, the exposure covariate is the location of the recommended site. We replace the original simple sigmoid function model that gets the result by dot product x of the exposure covariate with a more complicated Neural Collaborative Filtering framework [16,17]. By doing this, we find that empirical performance is susceptible to this choice, motivating the need to place models on priors to obtain μ_{ui} with flexible parameters.

The observed exposure covariates x_i and exposure model parameters u and conditional coefficients $u_i|x_i$, ψ_u are introduced according to a domain-specific structure. The extended graphical model with the exposure covariate is shown previously in Figure 6. No matter the appearance of this exposure, the conditional independence between the exposed prior and the more standard collaborative filtering parameters (a given direction) will ensure that updates to our model are the same for many popular inference programs, turning the covariates-exposing extension into a plugin.

4. Optimization based on Gaussian mixed model

Consider two sets of variables for every combination of users $u = 1,...,U$ and items $i = 1,...,I$. The first matrix $A = a_{ui}$ indicates whether user u has been exposed to item i. The second matrix $Y = y_{ui}$ means whether or not user u clicked on item i. The possibility of whether a user is exposed to an item obeys the Bernoulli distribution. As a condition of disclosure, the case the user u clicks on item i comes from a matrix factorization model. Similar to the standard methodology, we factorize this conditional distribution to M user preferences $\theta_{i,1:K}$ and M item attributes $\beta_{u,1:K}$

4.1. Model inference

$$\theta_u \sim \mathcal{N}(0, \lambda_\theta^{-1} I_k)$$
$$\beta_i \sim \mathcal{N}(0, \lambda_\beta^{-1} I_k)$$
$$a_{ui} \sim \text{Bernouli } (\alpha_{ui})$$
$$y_{ui} \mid a_{ui} = 1 \sim \sum_{k=1}^{n} \phi_k \cdot \mathcal{N}(\mu_k \theta_u^T \beta_i, \sigma_k^2)$$
$$y_{ui} \mid a_{usi} = 0 \sim \delta_0$$

where δ_0 denotes that $p(y_{ui} = 0|a_{ui} = 0) = 1$, and we introduced a set of hyperparameters indicating the inverse variance $(\lambda_\theta, \lambda_\beta)$. α_{ui} is the prior probability of exposure.

We observe matrix Y has a relationship with matrix A. There is a covariance between matrix Y and matrix A. When $y_{ui} = 1$, we know that $a_{ui} = 1$. That means if user *u* clicked item *I,* user *u* must have been exposed to item *i.* When $y_{ui} = 0$, a_{ui} is latent. There are two different reasons leading to this result. One is user *u* may have been exposed to item *i* and decided not to click (i.e., $a_{ui} = 1$, $y_{ui} = 0$); another is user u may have never seen item i(i.e., $a_{ui} = 0$, $y_{ui} = 0$). Since Y is usually sparse in practice, most a_{ui} will be potential.

Based on the Exposure MF model [Dawen Liang ,2017],I try to do further work on the original model.In the Exposure MF model, the author assumed that $y_{ui}|a_{ui} = 1 \sim \mathrm{N}(\theta_u{}^T \beta_i, \lambda{-}_y{}^1)$. Compared with this Gaussian model, the Gaussian mixed model will be more suitable for a dataset with more uncertainties.

We continue our work based on Gaussian Mixed Model.

Now we have $y_{ui} \mid a_{ui} = 1 \sim \sum_{k=1}^{K} \phi_k \cdot \mathcal{N}(\mu_k \theta_u^T \beta_i, \sigma_k^2)$. We then define the probability that user u click item i when the item i is exposed to user i . According to the Gaussian mixed model, the likelihood that can be calculated according to

$$P(yui|\theta u, \beta i, \phi, \mu, \sigma \mathbf{2}, aui = 1) = \sum_{k=1}^{k} \phi k \cdot \mathrm{N}(yui|\mu k, \theta u, \beta i, \sigma k2, aui = 1) \quad (2)$$

For the sake of convenience, we define two sets of sample indices:

$$I1 = \{(u,i)|y(u,i) = 1, (u,i) \in U \times I\} \quad (3)$$
$$I0 = \{(u,i)|y(u,i) = 0, (u,i) \in U \times I\} \quad (4)$$

We respectively refer to I_1 and I_0 as the positive sample set and negative sample set.

Given parameters $\Theta = \{\mu, \sigma, \theta_u, \beta_i, \phi\}$, the likelihood of observation $D = \{y_{ui}\}_{(u,i)}$ can be factorized as

$$p(\mathcal{D};\theta) = \prod_{(u,i)} \sum_{a_{ui} \in \{0,1\}} p(a_{ui} \mid \theta_u, \beta_i, \pi, \mu, \sigma^2) p(y_{ui} \mid \theta_u, \beta_i, \pi, \mu, \sigma^2, a_{ui}) \quad (5)$$

Note that if $a_{ui} = 0$, $y_{ui} = 0$ must be true. The equations are thus satisfied consistently.

$$p(yui = 0|aui = 0, \theta u, \beta i, \pi, \mu, \sigma 2) = 1 \quad (6)$$

$$p(yui = 1|aui = 0, \theta u, \beta i, \pi, \mu, \sigma 2) = 0 \quad (7)$$

if $y_{ui} = 1$,$a_{ui} = 1$must be true,

$$p(aui = 1|yui = 1, \theta u, \beta i, \pi, \mu, \sigma 2) = 1 \quad (8)$$
$$p(aui = 0|aui = 1, \theta u, \beta i, \pi, \mu, \sigma 2) = 0 \quad (9)$$

Therefore, by plugging (8) and (9) into likelihood (5) and dividing the sample indices into two sets I_1 and I_0, the likelihood can be deformed as

DOI: 10.54254/2755-2721/6/20230929

$$p(\mathcal{D};\theta)=\prod_{(u,i)\in I_1} p(\,a_{ui}=1\mid\alpha_{ui}\,)\cdot p(\,y_{ui}=1\mid a_{ui}=1,\theta_u,\beta_i,\pi,\mu,\sigma^2\,)$$
$$\times\prod_{(u,i)\in I_0 a_{ui}=\{0,1\}} p(y_{ui}=0\mid a_{ui},\theta_u,\beta_i,\pi,\mu,\sigma^2)\cdot p(a_{ui}\mid\alpha_{ui})$$

Table 1. Notations.

Variable	Description
$U=\{1,2,3,\ldots,U\}$ $I=\{1,2,3,\ldots,I\}$	Cartesian product consisting of the set of users and the set of items
$(u,i)\in U\times I$ $Y=\{y_{ui}\}\in\{0,1\}$	Whether user u have clicked item i
$A=\{a_{ui}\}\in\{0,1\}$	Whether item i has been exposed to user u
$\alpha=\{\alpha ui\}$	The exposure rate for item i to user u
K	Number of mixed components
$\theta u\in RM$	Latent preferences vector of user u ,where k is the length of the feature vector
$\lambda i\in RM$	Latent attributes vector of item i , where k is the length of the feature vector
$\phi=(\phi 1\ldots\phi K)$	K-dimensional vector composed of all the individual $\phi_{1..K}$, must sum to 1.$\phi_{1..K}$ indicates the mixture weights,i.e., the prior probability of a particular component k.
$\mu=(\mu 1\ldots\mu K)$	K-dimensional vector composed of all the individual $\mu_{1..K}$, indicates the mean of a particular component k.
$\boldsymbol{\sigma^2}=(\sigma_1^2\ldots\sigma_K^2)$	K-dimensional vector, composed of all the individual$\sigma_{1..K}^2$s, indicates the variance of a particular component k.

4.2. Learning algorithms

This section describes the GMM learning algorithm based on the EM algorithm. We first define the objective function in the standard way of the EM algorithm.

According to Jensen's inequality, a lower bound $Q(\Theta;\Theta)^{-}$ for the logarithm of likelihood (14) can be derived as

$\log p(D;\Theta)\geq Q(\Theta,\Theta)^{-}=$

$$\sum_{(u,i)\in\mathcal{I}_1}\log\, p(a_{ui}=1\mid\alpha_{ui})\cdot p(y_{ui}=1\mid a_{ui}=1,\theta_{u,},\beta_i,\pi,\mu,\sigma^2)]$$

$$+\sum_{(u,i)\in\mathbb{I}_0}\sum_{a_{ui}=\{0,1\}}\bar{q_{uia}}\log\,[P(y_{ui}=0\mid a_{ui},\theta_u,\beta_i,\pi,\mu,\sigma^2)p(a_{ui}\mid a_{ui})] \tag{11}$$

where q_{uia}^{-} is a posterior of a_{ui}

$$q_{uia}^{-}=P(a_{ui}=c|\theta_u,\beta_i,\pi,\mu,\sigma^2,\alpha_{ui},y_{ui}=0)$$

and the total value of q_{uia}^{-} should equal to 1, which satisfies

$_{ui}\sum q_{uia}^{-}=1\ a\in 0,1$ We can get the q_{uia}^{-} in the E step:

$$q\bar{ui}1=p(aui=1|\theta u,\beta i,\pi,\mu,\sigma 2,\alpha ui,yui=0)$$
$$=E(aui|\theta u,\beta i,\pi,\mu,\sigma 2,\alpha ui,yui=0)$$
$$=\frac{p(a_{ui}=1,y_{ui}=0\mid\theta_u,\beta_i,\pi_k,\mu_k,\sigma_k^2,\alpha_{ui})}{p(a_{uj}=1,y_{ui}=0\mid\theta_{u,},\beta_i,\pi,\mu,\sigma^2,\alpha_{ui})+p(a_{ui}=0,y_{uj}=0\mid\theta_{u,},\beta_i,\pi,\mu,\sigma^2,\alpha_{ui})}$$

DOI: 10.54254/2755-2721/6/20230929

$$= \frac{p(a_{ui} = 1 \mid \alpha_{ui}) \cdot p(y_{ui} = 0 \mid \theta_u, \beta_i, \pi, \mu, \sigma^2, a_{ui} = 1)}{p(a_{ui} = 1 \mid \alpha_{ui}) \cdot p(y_{ui} = 0 \mid \theta_u, \beta_i, \pi, \mu, \sigma^2, a_{ui} = 1) + p(a_{ui} = 0 \mid \alpha_{ui}) \cdot p(y_{ui} = 0 \mid \theta_u, \beta_i, \pi, \mu, \sigma^2, a_{ui} = 0)}$$

$$= \frac{\alpha_{ui} \cdot \sum_{k=1}^{K} \phi_k \cdot N(y_{ui} \mid \mu_k, \theta_u, \beta_i, \sigma_k^2)}{\alpha_{ui} \cdot \sum_{k=1}^{K} \phi_k \cdot N(y_{ui} \mid \mu_k, \theta_u, \beta_i, \sigma_k^2) + (1 - \alpha_{ui})}$$

We can get the value of $\bar{q}_{ui0}$ from the equation: $\bar{q}_{ui0} = 1 - \bar{q}_{ui1}$ 。The E-step is to update the posterior of a_{ui} under current estimates of parameters Θ. Then in the M-step, we update the parameters Θ by employing a gradient-based optimization method such as the gradient descent method or quasi-Newton method, fixing the posterior of a_{ui}. We add l_2 regularizers for(θ ,β),to (11), which are respectively controlled by strength parameters λ_θ, λ_β. For the parameters of the Gaussian mixed model, we just use the maximum likelihood to update (ϕ,μ,σ^2). When updating the parameter ϕ_k, we need to add the Lagrange multiplier because $\sum_{h=1}^{K} \phi_h = 1$.

By alternatively conducting E-steps and M-steps until converging the objective function (11), we iteratively estimate the parameters Θ. The learning algorithm for GMM is shown in Algorithm 1 in the summary. In the algorithm, lbfgs_update_ θ and lbfgs_update_β are functions that update θ and β through using L-BFGS, and the arguments are the initial values of the parameters and the gradients of the parameters.

We get this algorithm.

Algorithm 1 Parameter estimation of ExpoMF by EM-algorithm
Input: training set D,maximum iterations $N \in \mathbb{N}^+$, tolerance $\epsilon > 0$.
//Initialization of Parameter n
$\theta_u \sim N(0,\lambda_\theta^{-1} I_k)$ $\beta_i \sim N(0,\lambda_\beta^{-1} I_k)$
$\alpha_{ui} \in [0,1]$ $\mu_k \in [0,\lambda\{_\mu]$
$\sigma_k^2 \in [0,\lambda^2_\sigma]$
$\pi_k \in [0,1)$ $\sum_{k=1}^{K} \pi_k = 1$
Θ(0) = μ0,σ0,θ**0**,β**0**,π**0**}
// Parameter estimation **for** j = 1, 2, · · · ,N **for**
// E-step computation **for** $(u,i) \in I_0$ **for**
for $a_{ui} \in \{0,1\}$

$$\bar{q}_{ui1} = \frac{\alpha_{ui} \cdot \sum_{h=1} \pi_h \cdot N(y_{ui} \mid \mu_h, \theta_{u,}, \beta_i, \sigma_h^2)}{\alpha_{ui} \cdot \sum_{h=1}^{K} \pi_k \cdot N(y_{ui} \mid \mu_k, \theta_{us}, \beta_i, \sigma_h^2) + (1 - \alpha_{ui})}$$
$$\bar{q}_{ui0} = 1 - \bar{q}_{ui1}$$

end
for
End for // M-step computation θ_u^j =lbfgs_update_theta $(\theta_i^{j-1}, \frac{\partial}{\partial \theta_u} Q(\Theta; \overline{\theta}^{(j-1)})$
β_i^j = lbfgs_update_beta $(\beta^{j-1}, \frac{\partial}{\partial \beta_k} Q(\theta; \overline{\theta}^{(j-1)})$
μj = argmax$_\mu$Q(Θ;$\bar{\Theta}$(j−1))
σj = argmaxσQ(Θ;$\bar{\Theta}$(j−1))
// When update π,we need to use the Lagrange multiplier π^j = argmax$_\pi$Q(Θ; $\Theta^{(j-1)}) + (\lambda \cdot \sum_{k=1}^{K} \pi_k - 1)$
$\Theta^j = \{\mu^j, \sigma^j, \theta^j, \beta^j, \pi^j\}$
// Checking stop condition **if**Q(Θ(j);Θ(j-1) − Q(Θ(j-1);Θ(j-2) ϵ **then**
break
end if
end for
Output: $\Theta^{(j)}$

DOI: 10.54254/2755-2721/6/20230929

5. Empirical study

In this section, we investigate the performance of the new ExpoMF++ model by fitting the model to multiple datasets. By exploring the obtained model works, we further understand the version of the optimized ExpoMF.

The optimized ExpoMF++ outperforms the original ExpoMF on four datasets representing user clicks, check-ins, bookmarks, and listening behaviors.

We used a mixture of GMF, MLP, and NeuMF for modeling, and the comparisons with previous methods and unoptimized ExpoMF are made through multiple metrics.

Through the tests, we found that when using GMF, MLP, NeuMF to optimize the exposure prior probability u_i, its performance is further improved. Compared with the original ExpoMF, adding GMF, MLP, and NeuMF has greatly enhanced the new algorithm's performance. Both the ExpoMF++ with location covariate and ExpoMF with content covariate outperform the simple ExpoMF with every *i* per item. We conduct comprehensive experiments concerning the following three key topics:

1. Under the same conditions, is our optimized ExpoMF++ better than the original ExpoMF onvarious indicators or those baselines?
2. Parameter Analysis and Ablation Studies.
3. Analysis characteristics of the proposed method during training.

5.1. Datasets in experiments

Gowalla: A dataset containing user location check-ins from location-based social networks. The data is processed to guarantee all users and venues have at least 20 check-ins. Additionally, the dataset comprises venue locations to guide location-based recommendations.

Taste Profile Subset (TPS): It contains user song play counts collected by music intelligence company Echo Nest.7 We interpret the play count as an implicit preference by binarizing it. We further preprocess the dataset to keep users who have at least 20 songs in their listening history and songs that were listened to by at least 50 users.

5.2. Experimental setup

The observed user-project interaction is randomly divided s into train/test/validation sets in a 70/20/10 ratio for each dataset. In all experiments, the potential spatial dimension of the co-filtering model K is 100. The model is trained according to the inference algorithm proposed above. Truncated normalized discounted cumulative gain is used on the validation set to monitor and ensure the algorithm's convergence. Same guidelines are applied to choose hyperparameters for ExpoMF++-based models and baseline models. For predictions, for each user *u*, we rank each item *i* by multiplying vectors. We then exclude artifacts from the training and validation sets and calculate all metrics from the resulting ordered list.

Furthermore, when using ExpoMF++ with exposure covariates, we found that NeuMF, GMF, and MLP performance were all improved to varying degrees by predicting missingness preference according to $E[y_{ui}|u,i]$ (see Section 5.4 for details).

5.3. Performance measures

Recall@k: For each user *u*, Recall@k is the recall rate -- the ratio of the number of relevant results retrieved in the previous top *K* results to the number of all relevant developments in the library of the retrieval system.

MAP@k: MAP is short for Mean Average Precision, and it calculates the mean of users' average precision, sensitive to the given order. For example, we query a keyword in a search engine, and 10 results are returned. Of course, the best-case scenario is that these 10 results are all relevant information. But if only a portion of them is appropriate, such as 5, then these 5 results are relatively good if displayed relatively early, and worse if they are placed starting from the 6th. This is the indicator reflected by average precision, which is somewhat similar to the concept of recall rate but is order sensitive.

DOI: 10.54254/2755-2721/6/20230929

NDCG@k: This variable emphasizes the significance of the top ranks by logarithmic discounting classes. In the recommendation system, CG, meaning "Cumulative Gain", represents the accumulation of the relevance scores of each recommendation result as the score of the entire recommendation list. However, CG does not consider the impact of the different positions of each result on the unqualified recommendation. For example, we always hope that the results with a significant correlation are ranked closer to the front, the low relevance ranking will affect the user experience. CG disregards the order, as unlike MAP it is not order-sensitive.

Preprocessed dataset properties: Referring to the density (Y) of the user-item consumption matrix, interactions are non-zero items (listen to counts, clicks, and check-ins),

Table 2. Attributes of Datasets after pre-processing.

Dataset	TPS	Gowalla
# of users	221830	57629
# of items	22781	47198
# interactions	14.0M	2.3M
% interactions	0.29%	0.09%

5.4. Evaluation of ExpoMF++ on various metrics

To evaluate the recommendation performance, we use Recall@k(the standard information retrieval measure) and two ranking-specific metrics(mean average precision (MAP@k) and NDCG@k)to conduct the analysis. We used a mixture of GMF, MLP, and NeuMF for modeling, and the comparisons with previous methods and unoptimized ExpoMF are made through multiple metrics. When using ExpoMF++ combined with exposure covariates, we found that NeuMF, GMF, and MLP performance were all improved in varying scales. The results are shown in the table below.

ExpoMF++ build close ties between the exposure of the user and the popularity of the item, partly due to the reason that the model's prior is parameterized by the term *i* of each item. By using exposure covariates, we can provide additional information about a user’s (potential) exposure to an item.

Table 3. Comparison between WMF, ExpoMF and ExpoMF++ with NeuMF method.

While the differences in performance are generally minor, ExpoMF++ performs comparably better than ExpoMF and WMF across datasets.

	Gowalla					TPS				
	WMF	ExpoMF	ExpoMF++	Location ExpoMF	Location ExpoMF++	WMF	ExpoMF	ExpoMF+	Location ExpoMF	Location ExpoMF++
Recall@20	0.122	0.118	0.121	0.129	0.132	0.194	0.200	0.206	0.202	0.209
Recall@50	0.192	0.186	0.194	0.199	0.202	0.293	0.284	0.293	0.289	0.295
NDCG@100	0.118	0.116	0.118	0.125	0.127	0.252	0.263	0.267	0.268	0.270
MAP@100	0.044	0.043	0.045	0.048	0.05	0.089	0.108	0.117	0.113	0.120

Recall that ExpoMF++ does what these exposure covariates do to allow the matrix factorization component to focus on items exposed to the user. In the model, this can be achieved by increasing the weight of items that users have likely been exposed to, increasing the probability of exposure, and down-weighting things that users have not been exposed to.

DOI: 10.54254/2755-2721/6/20230929

Table 4. Comparison between WMF, ExpoMF, and ExpoMF++ with GMF method.

While the differences in performance are generally minor, ExpoMF++ performs comparably better than ExpoMF and WMF across datasets. ExpoMF++ outperforms the simpler ExpoMF and WMF according to all metrics using location exposure covariates.

	Gowalla					TPS				
	WMF	ExpoMF	ExpoMF++	Location ExpoMF	Location ExpoMF++	WMF	ExpoMF	ExpoMF+	Location ExpoMF	Location ExpoMF++
Recall@20	0.122	0.118	0.120	0.129	0.131	0.194	0.200	0.202	0.202	0.205
Recall@50	0.192	0.186	0.193	0.199	0.201	0.293	0.284	0.293	0.289	0.293
NDCG@100	0.118	0.116	0.118	0.125	0.127	0.252	0.263	0.264	0.268	0.269
MAP@100	0.044	0.043	0.044	0.048	0.048	0.089	0.108	0.110	0.113	0.115

Table 5. Comparison between WMF, ExpoMF, and ExpoMF++ with MLP method.

While the differences in performance are generally minor, ExpoMF++ performs comparably better than ExpoMF and WMF. By combining linear GMF and nonlinear MLP and using location exposure covariates,ExpoMF ++ outperforms the simpler ExpoMF and WMF according to all metrics.

	Gowalla					TPS				
	WMF	ExpoMF	ExpoMF++	Location ExpoMF	Location ExpoMF++	WMF	ExpoMF	ExpoMF+	Location ExpoMF	Location ExpoMF++
Recall@20	0.122	0.118	0.119	0.129	0.130	0.194	0.200	0.204	0.202	0.207
Recall@50	0.192	0.186	0.191	0.199	0.200	0.293	0.284	0.293	0.289	0.294
NDCG@100	0.118	0.116	0.117	0.125	0.125	0.252	0.263	0.265	0.268	0.269
MAP@100	0.044	0.043	0.044	0.048	0.049	0.089	0.108	0.111	0.113	0.116

Through data analysis and training, we found that:

1.Compared with the original model and other baseline methods, the ExpoMF++ model has greatly improved under the evaluation indicators of Recall@k and NDCG@k.

2.The ExpoMF++ model framework works better when using the NeuMF effect, better than GMF and MLP

3.ExpoMF++ model with NeuMF without pre-training requires more epochs to get good results.

4.ExpoMF++ model with GMF outperforms ExpoMF++ model with MLP, and the larger the dataset, the better it works.

5.5. Model analysis

In this section, we will analyze our proposed method thoroughly in order to help readers better understand its nature and the properties of the proposed new model.

Table 6. Parametric Analysis.

variable	definition
epochs	Number of training rounds
m	Negative sampling rate
η	SGD learning rate
d	Embedding dimensions
std	The standard deviation of the model parameters (standard normal distribution)
λ	Regularization parameter

We use grid search to control the learning rate $\eta \in \{0.001,0.003,0.01\}$ and negative sampling rate $m = \{4,8,16\}$ for the first round of rough sampling. We show some crucial variables in the table. We then choose the optimal ones to begin the finer selection in the second round with $\lambda = \{0.001,0.003,0.01\}$. At the same time, we fix the number of epochs, embedding dimensions, and standard deviation.

Compared with the original model and other baseline methods, the ExpoMF++ model has dramatically improved under the evaluation indicators of Recall@k and NDCG@k. The model framework works better when using the NeuMF effect, better than GMF and MLP when combining linear GMF and nonlinear MLP. But NeuMF without pre-training requires more epochs to get good results.

Exposure-based neural network models are interpretable and learn flexible weights. By optimizing marginal probabilities, the model can adaptively learn the exposure probabilities and convert them into confidence weights to compensate for exposure biases. Using the NeuralCF network instead of the original exposure level modeling, it can better iteratively model the user-item model without changing the complexity of the original algorithm and improve the accuracy of the predicted value to improve the performance of the original model.

6. Related work

This section highlights the relationship between the ExpoMF++ model and other similar research results.

6.1. Causal inference

Usually, we cannot observe both the results of the intervention and the results of non-intervention in the same research project. For the research subjects who received the intervention, the state where they did not receive the intervention was a" counterfactual" state; for those who did not receive the intervention, the state where they received the intervention was also a" counterfactual" state.

So, we use causal inference [18] to determine whether the user click-through rate will increase with the emergence of new recommendation systems.

From the perspective of causal inference, specific characteristics of the user, such as age, region, and preference, may affect not only the exposure of the item to the user but also the user's feedback on the exposed item. Our model exploits the fork structure on the causal graph. We can understand exposure as a latent variable from the perspective of causal analysis, so the exposure process of our modeling items is modeling the causal process.

The personalized recommendation problem we are interested in varies from traditional causal inference work, which aims to explore the actual causal relationship between things and remove those confounding pseudo-causal relationships. By combining with the latent outcome framework, we model the latent variables as generative models with the logic of probability graphs.

6.2. Neural Collaborative Filtering

The NeuralCF (Neural Collaborative Filtering) network involved can be decomposed into two subnetworks: the Generalized Matrix Factorization and the Multi-Layer Perceptron. Both sub-networks contain representations of users and items. The overall network is optimized by training GMF and MLP separately and loading their weights.

In the base model, exposure is presented hierarchically. One way is to model according to the popularity of the item. The higher the rage, the higher the exposure rate. It does this by selecting a conjugate prior model related to the thing that does not use external information.

The other way is the context-based prior probability of u_i, that is, using the p_{ui} generated in E-step before supervising u_i. We use the NeuralCF network to replace the original exposure level modeling. The prior u_i is estimated by GMF and MLP [19,20], which are trained separately. By using linear kernel function, GMF models the potential feature interaction, and MLP [21,22] uses a nonlinear kernel function to learn the interaction function from the data, so we use the NCF framework to fuse GMF and MLP to enhance each other, better understanding the complex data provided.

The user-item matrix iterative interaction for modeling improves the accuracy and overall performance of μ_{ui} without changing the algorithm's complexity.

6.3. Versatile CF models

CF (Collaborative Filtering) models are generally specialized for modeling a single type of data (eg, document content when recommending scientific papers [citations]). An exception is Rendle [quote]'s Decomposition Machine (FM) [23]. FM can model all types of (numerical) user, item, or user-item characteristics by considering interactions between all features. Then FM can help learn specific parameters for each interaction.

In contrast, the exposure matrix of our ExpoMF++ model can be used to model external forms by describing user and item interactions, which is in contrast to most CF models. The models, though seemingly conflicting with each other, are not limited to single data modeling.

7. Conclusion

Exposure bias limits recommendation systems by masking both unfavored and underexposed items under the "not-clicked" zero value, but we have a solution for that.

Exposure Matrix Factorization Plus (abbreviatedas ExpoMF++), our optimizationmodel for a recommendation based on the basic Exposure Matrix Factorization (ExpoMF), differs from the original in two ways. In short, compared with the original model, the ExpoMF++ model has two aspects of optimization and reconstruction.

First, Neural Collaborative Filtering replaces the simple dot product function in the layers to obtain and update data. Second, we added an optimization to the distribution with the Gaussian mixture model. In our semi-reality tests, we found that our improvements positively affected reducing the exposure bias in recommendation systems. We also conducted many real-life data experiments in various datasets of different sizes to compare and prove that the ExpoMF++ model has dramatically reduced exposure bias issues compared with the original model and other baseline methods.

8. Future works

In the future, we will look forward to updating the algorithm with faster, more efficient methods previously undiscovered or from other aspects. We can modify them to fit the case of Exposure Bias. We will also try out the newly improved models in real-life instances, such as in custom websites and shopping applications, where customers and viewers will be exposed to many options.

Whatever research we may continue, one thing is sure: although exposure bias may never be fully cleared and might always hinder the vision of recommendation systems; reducing it - and developing the algorithms that do that - will pave our way through the enormous amount of information lying ahead of us in the infinite possibilities of the Internet.

Acknowledgment

Yizhao Chen and Han Li contributed equally to this work and should be considered co-first authors.

We have received great assistance while researching and writing the thesis. We would like to thank our supervisor, Professor David Woodruff, whose expertise in formulating research questions and whose methods have been invaluable to our studies.

References

[1] Chen J, Dong H, Wang X, et al. Bias and debias in recommender system: A survey and future directions[J]. arXiv preprint arXiv:2010.03240, 2020.
[2] Bonner S, Vasile F. Causal embeddings for recommendation[C] //Proceedings of the 12th ACM conference on recommender systems. 2018: 104-112.
[3] Liang D, Charlin L, McInerney J,etal.Modelinguserexposureinrecommendation[C].Proceedings of the 25th international conference on World Wide Web. 2016: 951-961.
[4] He X, Liao L, Zhang H, etal. Neuralcollaborativefiltering[C]//Proceedingsofthe26thinternational conference on world wide web. 2017: 173-182.
[5] Liu D, Cheng P, Zhu H, et al. Mitigating confounding bias in recommendation via information bottleneck[C]//Fifteenth ACM Conference on Recommender Systems. 2021: 351-360.
[6] Rasmussen C. The infinite Gaussian mixture model[J]. Advances in neural information processing systems, 1999, 12.
[7] McLachlan G J, Rathnayake S. On the number of components in a Gaussian mixture model[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2014, 4(5): 341-355.
[8] Zivkovic Z. Improved adaptive Gaussian mixture model for background subtraction[C] Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. IEEE, 2004, 2: 2831.
[9] Hajlaoui R, Guyennet H, Moulahi T. A survey on heuristic-based routing methods in vehicular ad-hoc network: Technical challenges and future trends[J]. IEEE Sensors Journal, 2016, 16(17): 6782-6792.
[10] Vanderhaegen F. Heuristic-based method for conflict discovery of shared control between humans and autonomous systems-A driving automation case study[J]. Robotics and Autonomous Systems, 2021, 146: 103867.
[11] Maashi M, Özcan E, Kendall G. A multi-objective hyper-heuristic based on choice function[J]. Expert Systems with Applications, 2014, 41(9): 4475-4493.
[12] Holland P W. Statistics and causal inference[J]. Journal of the American statistical Association, 1986, 81(396): 945-960.
[13] Peters J, Janzing D, Schölkopf B. Elements of causal inference: foundations and learning algorithms[M]. The MIT Press, 2017.
[14] Qian X, Feng H, Zhao G, et al. Personalized recommendation combining user interest and social circle[J]. IEEE transactions on knowledge and data engineering, 2013, 26(7): 1763-1777.
[15] Wang C, Zheng Y, Jiang J, et al. Toward privacy-preserving personalized recommendation services[J]. Engineering, 2018, 4(1): 21-28.
[16] He X, Du X, Wang X, et al. Outer product-based neural collaborative filtering[J]. arXiv preprint arXiv:1808.03912, 2018.
[17] Gao H, Xu Y, Yin Y, et al. Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services[J]. IEEE Internet of Things Journal, 2019, 7(5): 4532-4542.
[18] Dowe P. Causal processes[J]. Stanford encyclopedia of philosophy, 2008.
[19] Karlik B, Olgac A V. Performance analysis of various activation functions in generalized MLP architectures of neural networks[J]. International Journal of Artificial Intelligence and Expert Systems, 2011, 1(4): 111-122.
[20] Zhang C, Pan X, Li H, et al. A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2018, 140: 133144.
[21] Rendle S. Factorization machines[C]//2010 IEEE International conference on data mining. IEEE, 2010: 995-1000.
[22] Desai M, Shah M. An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN)[J]. Clinical eHealth, 2021, 4: 1-11.

DOI: 10.54254/2755-2721/6/20230929

[23] Zhou F, Zhou H, Yang Z, et al. EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction[J]. Expert Systems with Applications, 2019, 115: 136-151.