Constrained User Exposure Matrix Factorization in Recommendation System

Zhi-cai HUANG, Shun-zhi ZHU*, Weng WEI, Ying ZHONG and Nan QIN

Xiamen University of Technology Xiamen, 361024, P. R. China

*Corresponding author

Keywords: Recommendation system, Constrained matrix, Matrix factorization.

Abstract. Due to the massive items and users limited scope, many existing recommendation systems suffer a more and more serious sparse problem on rating matrix. Most of the recommendation systems employ the content-based method or the collaborative filter model to abbreviate the sparse problem. While in our paper, based on the assumption that users who have rated similar sets of items are likely to have similar preference information, we present a constrained user exposure matrix factorization collaborative filter model and apply it to deal with the matrix sparsity problem, which it makes a better recommendation to users who have very few ratings. In detail, for the sake of capturing the effect of user having rated a particular item, we introduce a constrained matrix to user latent vector space. This effect will have an impact on the prior mean of user latent feature vector. After that, using the user feature vector and item feature vector to estimate the values of unrated items. The experimental results show that the proposed model gets higher prediction precision than the state-of-the-art models in providing recommendation for user having very few ratings.

Introduction

Recommendation systems provide recommendation list by utilizing the explicit feedback data, which users have the specific rated value for items, but this kind of data are extremely sparse. Another recommendation approach is replacing the explicit feedback data with the implicit feedback data, which the implicit feedback consists of binary data reflecting a user’s behavior, such as the user exposure. However the non-positive sample in implicit feedback is ambiguous because of negative examples and unlabeled positive examples are mixed together and it is difficult to distinguish them.

To alleviate the sparse problem. Pan R et al. [1] propose the weighted matrix factorization method (WMF), which it use a simple heuristics where all the non-positive examples are equally downweighted with regard to the positive examples. But the WMF model is not a generative model, and it cannot deal with the situation that contains the latent variable. Dawen et al. [2] propose a generative model called Exposure matrix factorization model (ExpoMF), which adding an user exposure variable and considering another items information to influence the prior of user exposure variable. The experimental results of ExpoMF show that it gets a remarkable performance than other baseline. Nevertheless, the ExpoMF model still flaw to improve the sparse problem, i.e. it hard to make a better recommendation for casual users (casual users means who having few ratings). Once the ExpoMF model is fitted, its casual user feature vector tends to the average user feature vector. In order to solve this problem, Salakhutdinov et al. [3] introduce an additional way of constraining user-specific feature vectors to the global user feature space. Inspiring it, we propose a constrained user exposure matrix factorization model (CUEMF) based on the ExpoMF model, which it introduces a constrained matrix W to user latent feature space. Applying the constrained technique, our proposed model can capture the effect of a user that have rated a particular item, and this effect will impact on the prior mean of user’s feature vector. Experimental results show that the CUEMF model improves the recommendation accuracy for casual users compared with baseline.

The remainder of this article is organized as follows. In Section 2, we briefly introduce the latent factor model of collaborative filtering recommendation system and explain the related work in detail. In section 3 we formally introduce our proposed CUEMF model and optimization process in detail. In
section 4, we conduct experiments on Gowalla dataset. Finally, Section 5 concludes and discusses a few directions as future work.

**Relative Work**

Generally speaking, the latent factor model utilizes matrix factorization [4] to find out the relationship between users and items. It maps high-dimensional user behavior matrix to low-dimensional users and items latent factor space and make them directly comparable. The observed rating matrix can be the consumption matrix or the rating matrix. It can express as following formula: 

\[ \text{Noise} = y_{ui} - \theta_u^T \beta_i, \]

where \( \theta_u \) and \( \beta_i \) represented user \( u \)'s latent preferences and item \( i \)'s latent attribute, \( y_{ui} \) represents the observed rating of user-item rating matrix. In addition, the distribution of \( \theta_u \), \( \beta_i \) and \( y_{ui} \) are the same as \( \text{Noise} \)'s distribution, such as Gaussian matrix factorization [5].

Based on the Gaussian matrix factorization and in the need of dig data computation, Salakhutdinov et al. [3] propose a probabilistic matrix factorization recommendation (PMF) by mapping the high-dimensional similar matrix into different low-dimensional latent feature vector. Then both latent variable can be obtained by maximizing the log of the posterior. Due to the dimension of latent feature vector is relatively low, the PMF employs the gradient descent to infer both latent feature vector. The experiment result of PMF indicates that it can deal with large amount of data effectively and achieve high-precision. However, the PMF model have a defect, which the PMF model assume that only few factor impact on user preference, leading the PMF model treats the prior of casual user feature vector is closed to the prior mean [3]. Because all the PMF based methods hard to get a satisfied performance to make recommendation for casual users. Similarly, the ExpoMF model based on PMF model is difficult to get a good recommendation performance for casual users, which have poor performance in alleviating the sparsity problem. Accordingly, to solve the sparse problem of PMF model, Hao et al. [6] propose a novel social recommendation framework fusing a user-item rating matrix with the user’s social network using PMF model. It achieves significant performance to alleviate the data sparsity problem and can be extended to larger datasets. But it still ignores the fundamental flaws of PMF, thus Salakhutdinov devises a constrained PMF model, and the experimental results provide better recommendation improvement for casual users. Inspired of this idea, we proposed the CUEMF model evolved by the ExpoMF model. Concretely, the CUEMF model not only consider the exposure between user against item, but focus on improving the recommendation precision for casual users.

**Constrained User Exposure Matrix Factorization**

For user \( u=1,...,N \) and item \( i=1,...,M \), we construct a triple tuple \( (u,i,y_{ui}) \), which can be interpreted as user \( u \) rated item \( i \) with rating value \( y_{ui} \). In the Fig. 1, \( Y = \{y_{ui}\} \) represents the user-item rating matrix, which contains all the rating value \( y_{ui} \). The exposure matrix \( A = \{a_{ui}\} \) can indicate whether the user is exposed to the item or not. There lies two exposure statements for user against item, including user exposed to item and user not exposed to item. Consequently, the exposure matrix \( A \) follows the Bernoulli distribution. Beyond this, the latent user feature matrix \( \beta \in R^{KxN} \) (\( K \) is dimension of latent
space), the latent item matrix $\beta \in \mathbb{R}^{K \times M}$ and the constrained matrix $W \in \mathbb{R}^{K \times M}$ follow the Gaussian distribution. In formally, they can express as following.

$$
\begin{align*}
\gamma_u & \sim N(0, \lambda^{-1}_u I_k) \\
\beta_i & \sim N(0, \lambda^{-1}_i I_k) \\
w_i & \sim N(0, \lambda^{-1}_w I_k) \\
a_{ui} & \sim \text{Bernoulli}(\mu_{ui})
\end{align*}
$$

The exposure matrix $A$ have a special structure, concretely, $y_{ui} > 0$ equals to $a_{ui} = 1$, which stands for the user have been exposed to item. If $y_{ui} = 0$, id est $a_{ui}$ is latent, it exists in two situations: one is that the $a_{ui} = 1$ and $y_{ui} = 0$, meaning that the user is possible have seen the item but user does not click the item. What we need to point out here is that it is not sure if the user likes the item. Another situation is $a_{ui} = 0$ and $y_{ui} = 0$, which the user may have never seen the item, it can formula as $p(y_{ui} = 0 | a_{ui} = 0) = 1$. In addition, since exposure matrix $A$ is usually sparse in dataset, therefore, most $a_{ui}$ is latent. For learning the prior $\mu_{ui}$ of exposure matrix $A$, our model introduces observed exposure covariates $x_i$ and exposure model parameter $\varphi_u$, and our paper will give a detailed description about it in session 3.3. Furthermore, a set of hyper-parameters what denotes the inverse variances $(\lambda_y, \lambda_\beta, \lambda_r, \lambda_w)$ are introduced in our CUEMF model as well.

**Constrained User Exposure Matrix Factorization**

In order to improve the rating prediction performance for casual users, our model combines ExpoMF model with constrained part. It leads to the following log joint probability formula:

$$
\log p(a_{ui}, y_{ui} | \mu_{ui}, \gamma_u, \beta_i, W, \lambda^{-1}_i) = \log \text{Bernoulli}(a_{ui} | \mu_{ui}) + a_{ui} \log N(y_{ui} | (\gamma_u + \sum w_i^T a_{ui} \beta_i, \lambda^{-1}_i))
$$

(1)

Suppose that user not exposed to item, that is the $a_{ui} = 0$, our paper uses conditional probability of $a_{ui}$ what follows the prior of $\mu_{ui}$ to build the log joint probability. On the contrary, if the user is exposed to items, the log joint probability recovers standard matrix factorization, like WMF. Different from the user latent feature vector $\theta_u$, which described in related work. We replace the $\theta_u$ with the constrained part $\gamma_u + \sum a_{ui}$, where $a_{ui} \in \mathbb{R}^{1 \times M}$ represents $u^{th} (u = 1, \ldots, N)$ user exposure vector.

The constrained matrix $W$ can be interpreted as the distribution of user exposure on the latent K-dimension space. The $i^{th}$ ($i = 1, \ldots, M$) coloum of $W$ presents the accumulation effect a user who have rated some particular items, these effect will impact on the prior mean of user feature vector. As a result, users who have clicked the same (or similar) items or visit the same place will have similar prior distributions on user feature vector. So the $\sum a_{ui}$ can be interpreted as the mean of the prior distribution. The $\gamma_u$ can be seen as the offset added to the prior mean. Specifically, the $\theta_u$ and $\gamma_u$ could be equal when the prior mean is fixed at zero [3]. Therefore, for providing recommendation to casual users, unlike the traditional recommendation systems treat the latent feature factor of casual users tend to similar with the average user feature factor, the CUEMF model will put more concentration on what users are interested in.

**Exposure Covariates and Exposure Prior**

The CUEMF model learns the exposure covariates $x_i$ and the exposure prior $\mu_{ui}$ if there is specific extra information in original dataset, such as the content of a documents, the location of a venue. As show in Fig.1, there exists conditional independence between $\mu_{ui}$ and standard latent feature vector, which ensure that the updates for the CUEMF model will be the same for many popular inference
process. Concretely, for an L-dimensional vector \( x_i \), considering the text recommendation, it can be obtained from the content of document \( i \) with method of the natural language processing. While in domain of location-based recommendation, the cluster method is an effective way to find the expected assignment to \( L \) clusters for all the venues in the dataset of venue’s position. In both cases, exposure covariates \( x_i \) is normalized to 1. More formally, it can express as following.

\[
\mu_{ui} = \sigma(\phi_{ui}^T x_i)
\]

where \( \sigma \) is the sigmoid function, and the model parameter \( \phi_u \) can be learned for each user during inference process. In the location domain, \( \phi_u \) represents the user's interest points that are usually exposed to the geographical location. Take the location recommendation as an example, there is the \( i^{th} \) cluster in \( x_i \) standing for Beijing and \( x_{ui} \) is high. It can conclude that user \( u \) probably lives in or visits Beijing often. Therefore it would be the suitable choose to highlight the items in Beijing rather than other place.

**Model Learning**

**Update for Latent Feature Variable.** In order to optimize the log joint probability function, the optional solution is various, such as Markov chain Monte Carlo, variational inference or expectation-maximization. We choose expectation-maximization (EM) for reasons of efficiency and simplicity. The EM will find the maximum a posteriori estimates of unknown parameters of the model. For our CUEMF model, EM inference can figure out the optimal log likelihood of CUEMF model, more formally, in the E(xpectation)-step of EM inference it will get the expectations of the exposure of unknown data. And in the M(aximization)-step it will update the latent feature vector based on above expectation value as the maximum expectation value. The EM will inference alternating between the E-step and M-step until obtain an optimal result. From another aspect, EM inference is analytical for our model because it is conditional conjugate. It means that in our model the posterior distribution and the prior distribution of both latent feature variable are in the same family, which both latent feature variables follow Gaussian distribution. In detail, the Table 1 summarizes the procedure of EM inference.

<table>
<thead>
<tr>
<th>Table 1. Inference for CUEMF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: input: rating matrix ( Y ), exposure covariates ( x_{i,M} ) (( M ): number of items)</td>
</tr>
<tr>
<td>2: random initialization: user factors ( \gamma_{i,N} ) (( N ): number of users), item factors ( \beta_{i,M} ), constrained matrix ( w_{i,M} ), exposure prior ( \mu_{ui} ), and exposure model parameters ( \phi_N ).</td>
</tr>
<tr>
<td>3: while performance on validation set increases do</td>
</tr>
<tr>
<td>4: Compute expected exposure ( A )</td>
</tr>
<tr>
<td>5: Update user factors ( \gamma_{i,N} )</td>
</tr>
<tr>
<td>6: Update item factors ( \beta_{i,M} )</td>
</tr>
<tr>
<td>7: Update constrained matrix ( w_{i,M} )</td>
</tr>
<tr>
<td>8: Update the coefficients of exposure prior ( \phi_u )</td>
</tr>
<tr>
<td>9: end while</td>
</tr>
</tbody>
</table>

(1) Expectation step

As state earlier, \( y_{ui} > 0 \) means that \( a_{ui} = 1 \). Therefore, the E-step computes the expectation of the exposure latent variable \( E[a_{ui}] \) for \( y_{ui} = 0 \). The \( N(0, \sum\gamma a_{ui} + \sum a_{ui} \beta_i, \lambda^{-1}) \) represents the probability density function of \( N(\gamma a_{ui} + \sum a_{ui} \beta_i, \lambda^{-1}) \) evaluated at 0. In formally, it can express as following.
(3) Maximization step

In E-step, the \( E[a_{ui} | \mu_{ui}, \gamma_u, \beta_i, W, y_{ui} = 0] \) has computed. Later on, it will be used in M-step, for notational convenience, we denote \( q_{ui} = E[a_{ui} | \mu_{ui}, \gamma_u, \beta_i, W, y_{ui} = 0] \). Setting \( q_{ui} = 1 \) if \( y_{ui} = 1 \), in addition, the \( q_{ui} \) stands for the new \( a_{ui} \) after E-step. The update for the latent factors and constrained matrix can be expressed as following.

\[
\gamma_u \leftarrow \frac{\lambda \sum_i q_{ui} y_{ui} \beta_i - \lambda \sum_i q_{ui} \beta_i' \beta_i^T \sum_i q_{ui}}{\lambda \sum_i q_{ui} \beta_i' + \lambda \gamma I_k}
\]

\[
\beta_i \leftarrow \frac{\lambda \sum_i q_{ui} y_{ui} \gamma_u + \sum_i q_{ui} W q_{ui}^T \gamma_u (\gamma_u + \sum_i q_{ui})^{-1} + \lambda \beta_i}{\lambda \sum_i q_{ui} (\gamma_u + \sum_i q_{ui})^{-1} + \lambda \gamma I_k}
\]

where the \( I_k \) stands for the density matrix with the dimension \( K \). During the M-step, considering the complex of calculating the constraint matrix \( W \), we employ embarrassed parallel computation to get each \( w_i \).

\[
w_i \leftarrow \frac{\lambda \sum_i q_{ui}^2 \gamma_u - \lambda \sum_i q_{ui} \gamma_u \gamma_u - \lambda \sum_i q_{ui} \gamma_u \gamma_u (\gamma_u + \sum_i q_{ui})^{-1} + \lambda \beta_i}{\lambda \sum_i q_{ui} (\gamma_u + \sum_i q_{ui})^{-1} + \lambda \gamma I_k}
\]

**Update for Exposure Prior** \( \mu_{ui} \). In CUEMF model, the exposure priors \( \mu_{ui} \) and the others model variables (given \( \mu_{ui} \)) are conditional independence, which it ensures that the updates for the latent user factor and item factor and exposure variable are not altered. Depending on what we have described in session 3.2, in the domain of place recommendation the \( x_i \) represents the cluster of the place. Considering the computation complexity of the model, we resort to the mini-batch stochastic gradient descent (MBSGD) to get the \( \varphi_{ui}^{new} \).

\[
\varphi_{ui}^{new} \leftarrow \varphi_{ui} + \eta_i \nabla_i
\]

Where for some learning rate \( \eta_i \), the \( \nabla_i \) can define as:

\[
\nabla_i = \frac{1}{|B_i|} \sum_{i \in B_i} (q_{ui} - \sigma(\varphi_{ui}^T x_i)) x_i
\]

At each iteration \( t \), we randomly subsample a small batch of item \( B_t \) and upgrade the gradient variable \( \nabla_t \).

**Experiments**

**Dataset Construction**

In our study, using the Gowalla dataset [7] to training our model. The Gowalla dataset contains user-venue checksins from a location-based social network. It also contains the locations for the venue,
which it can use the location cluster information to obtain the exposure variable $x_j$. In order to make full use of the dataset, paper resort to 5-fold cross validation to pre-process the Gowalla dataset with 80% randomly selected ratings for training and the remaining 20% for testing.

**Evaluation Metric**

To measure the performance of different methods from multiple aspects, three popular used metrics are employed in our study, including Recall, Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). It is worth mentioning that $k$ presented the top-$k$ candidate items during our entire experiment.

First, for each user $u$ the Recall@$k$ computes as follow:

\[
\text{Recall}@k = \sum_{i \in \{ y_{u}^{\text{test}} \} \cap \Omega} \frac{\Omega\{ \text{rank}(u,i) \leq k \}}{\min(k,|y_{u}^{\text{test}}|)}
\]

where the $y_{u}^{\text{test}}$ is the set of rated items in the testdata for user $u$. The $\text{rank}(u,i)$ presents the rank of item $i$ in user $u$’s predicted list. The $\Omega\{\cdot\}$ is the indicator function. As we know the Recall and Precision are contradictory metrics, generally speaking, if the Recall gets a high value and the Precision would have a low value. Take an extreme instance, sampling a positive case from the overall sample randomly. As a consequence, the Precision of this sample will reach 100%, yet the Recall would obtain a very low value.

The MAP calculates the mean of user’s average precision. The formal expression of MAP for each user is following, which $n$ in the MAP stands for the sequence number of $k$.

\[
\text{MAP} @ k = \frac{\sum_{n=1}^{k} \text{Precision} @ n}{\min(n,|y_{u}^{\text{test}}|)}
\]

The last metric is NDCG, which calculates the top ranks by logarithmically discounting method. The expression of NDCG@$k$ for each user is as following.

\[
\text{DCG} @ k = \sum_{i=1}^{k} \frac{2^{a_i} - 1}{\log_i (i+1)}; \text{NDCG}@k = \frac{\text{CDG}@k}{\text{IDCG}@k}
\]

where the IDCG is the Ideal DCG, the IDCG calculates the DCG@$k$ after ranking the result list in a best permutation. It normalizes the DCG and ensures that the value of NDCG is between 0 and 1. The binary variable $Bel_i$ presents if the item $i$ belongs to $y_{u}^{\text{test}}$, which leading the $Bel_i=1$ if $i \in y_{u}^{\text{test}}$ and 0 otherwise.

**Details of Training**

In order to speeding up the training efficiency, our model utilizes the embarrassed parallel computation to update the latent variable and the constrained matrix. With this technique it can shorten the time of training model. In our experiment the batch size set to 1000 for the convenience of embarrassed parallel computation. Furthermore, the mini-batch SGD method is adopted here to update the exposure covariates. After trying various values, our model choses a learning rate of 0.1 and an update epoch of 10.

**Performance Compare between Three Models.** To demonstrate the different performance of ExpoMF and WMF model compared with CUEMF, we conduct them with above-mentioned metric. All of involved parameters are carefully turned on the same validation set, and the best performance of both methods is used to report the final comparison results. Table 2 summarizes the comparison results. It have the following observations: the WMF model performs well than the ExpoMF model in every metrics. As for both CUEMF and ExpoMF model, the CUEMF model achieves an improvement than ExpoMF and WMF in MAP@100 and NDCG@100. Because of the CUEMF model constrains the user’s latent factor and downweights user’s latent factor, therefore, it makes the recommendation item list of CUEMF tends to be more relevant to user’s interests. Arguably, CUEMF model make use of the constrained matrix to ensure that the recommendation systems make the user latent feature vector concentrated on what the user are interested in.
Table 2. Comparison results.

<table>
<thead>
<tr>
<th>Model</th>
<th>WMF</th>
<th>ExpoMF</th>
<th>CUEMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall@30</td>
<td>0.157</td>
<td>0.1572</td>
<td>0.1576</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recall@60</td>
<td>0.216</td>
<td>0.2158</td>
<td>0.2155</td>
</tr>
<tr>
<td>MAP@100</td>
<td>0.048</td>
<td>0.0478</td>
<td>0.0482</td>
</tr>
<tr>
<td>NDCG@100</td>
<td>0.125</td>
<td>0.1252</td>
<td>0.1255</td>
</tr>
</tbody>
</table>

Performance for Casual Users. In order to display the recommendation performance for casual users, the Gowalla testing data is divided into six groups by the number of checkins. Then the experiment runs on each observed checkin groups. During our experiment, all of the latent feature vectors are initialized to the same value in three models. Alternatively, all of the variance parameters set to the same value as well.

![Bar chart: distribution of users by the number of observed checkins in testing data, where the x-axis shows the number of observed checkins. Line chart: showing the performance difference of three models in NDCG.](image)

Note that the bar chart of Fig.2 displays the proportion of each group, the proportion of users who have 1-2 checkins is 39%, which take the majority in the test data, as for the others users 3-4, 5-8, 9-16, 17-20, more than 20 have the proportion of 23%, 20%, 11%, 2%, 5% respectively. In line chart of Fig.2, the curve of WMF@100 is in the lowest position compared with the ExpoMF’s and the CUEMF’s curve. Furthermore, it is observed that the CUEMF model outperforms better than the ExpoMF for users who have 1-2 checkins, the CUEMF achieves NDCG@100 of 0.0419 and ExpoMF achieves NDCG@100 about 0.0416, which it get about 0.7% better than ExpoMF model. As what we description in section 4.2, the NDCG is a metric of ranking quality and it is accumulated from the top to the bottom of the estimation list, that is to say the value of NDCG is pertinent to the number of user. As the user number drop, NDCG@100 of both model shows a significantly decline. In addition, it can be obtained from the diagram the performance of CUEMF is superior than the other models before checking number 9-16. However after this checkins regions, the ExpoMF shows better performance than the other models. Finally with the increase of checkins number, three models tend to have the same performance. From the experiment result, it indicates that the CUEMF model is able to generalize considerably better for casual users.
Summary
In this paper, we present a CUEMF model that contained a constrained matrix to user feature vector. After experiments, it shows the CUEMF outperforms on providing recommendation for casual users. However the time consumption of CUEMF is long, we plan to resort to the improved EM method to hoist the updated speed of latent feature variables. Furthermore, we hope that more information about venue or visit place can take into consider for the recommendation system, such as the property of this place, the service or the visitor flowrate.

Acknowledgement
This work is supported by the National Natural Science Foundation of China (NSFC.61672442, 61373147), the Science and Technology Planning Project of Fujian Province (No.2016Y0079).

References