Federated Probabilistic Preference Distribution Modelling with Compactness Co-Clustering for Privacy-Preserving Multi-Domain Recommendation

Weiming Liu\textsuperscript{1}, Chaochao Chen\textsuperscript{1*}, Xinting Liao\textsuperscript{1}, Mengling Hu\textsuperscript{1}, Jianwei Yin\textsuperscript{1}, Yanchao Tan\textsuperscript{2} and Longfei Zheng\textsuperscript{3}

\textsuperscript{1}College of Computer Science and Technology, Zhejiang University, Hangzhou, China
\textsuperscript{2}College of Computer and Data Science, Fuzhou University, Fuzhou, China
\textsuperscript{3}Ant Financial, Hangzhou, China
{21831010, zjuccc, xintingliao, humenling, zjuyjw}@zju.edu.cn, yctan@fzu.edu.cn, zlf206411@antfin.com

Abstract

With the development of modern internet techniques, Cross-Domain Recommendation (CDR) systems have been widely exploited for tackling the data-sparcity problem. Meanwhile most current CDR models assume that user-item interactions are accessible across different domains. However, such knowledge sharing process will break the privacy protection policy. In this paper, we focus on the Privacy-Preserving Multi-Domain Recommendation problem (PPMDR). The problem is challenging since different domains are sparse and heterogeneous with the privacy protection. To tackle the above issues, we propose Federated Probabilistic Preference Distribution Modelling (FPPDM). FPPDM includes two main components, i.e., local domain modelling component and global server aggregation component with federated learning strategy. The local domain modelling component aims to exploit user/item preference distributions using the rating information in the corresponding domain. The global server aggregation component is set to combine user characteristics across domains. To better extract semantic neighbors information among the users, we further provide compactness co-clustering strategy in FPPDM++ to cluster the users with similar characteristics. Our empirical studies on benchmark datasets demonstrate that FPPDM/FPPDM++ significantly outperforms the state-of-the-art models.

1 Introduction

Cross-domain recommendation (CDR) systems have shown great success in alleviating the serious data sparsity problem [Guo et al., 2021; Li et al., 2022; Zang et al., 2021a; Zheng et al., 2022]. It is well acknowledged that CDR models always share the user-item rating information across domains for joint collaborative filtering. Such methods can provide better performance on both source and target domains. However, with the increasing law on privacy protection, the user-item rating information are not accessible among different domains [Chen et al., 2022]. How to provide high-quality cross- and multi-domain recommendations under the demand of privacy protection has become an urgent problem.

In this paper, we focus on a more general problem of Privacy-Preserving Multi-Domain Recommendation (PPMDR). That is, user-item rating interactions are regarded as private information and other domains (participants) cannot directly obtain them. Such demand makes it difficult for knowledge transfer across domains to extract users’ domain-invariant preferences. Meanwhile, these multiple domains all suffer from the data sparsity problem. It further hinders the model to learning reliable representations and limits the model’s performance.

Existing research on CDR cannot solve the PPMDR problem well. Firstly, most conventional CDR models need to access the whole user-item rating interactions among different domains for modelling [Hu et al., 2018; Chen et al., 2020]. While it is not suitable under the settings of PPMDR, since the user-item ratings should be protected in each domain. Recently, FedMF [Chai et al., 2020] first adopts federated learning strategy for ensuring data privacy protection. However, FedMF only adopted a shallow model and thus cannot depict the complex user-item relationship well. Secondly, most of current CDR models only utilize embeddings rather than distributions to represent users and items, which leads to an inaccurate understanding of user preference, especially under the data sparsity scenario [Jiang et al., 2020; Ma and et al, 2020]. Specifically, we provide an example in Fig.1(a)-(b) where the user Mary prefers romantic and historical items while dislikes horror items. If we only consider the user/item embeddings, the horror items Suspect X or Kill Game will be recommended to Mary since they are closer than Woman in Love or Flipped. While when we model the users/items with Gaussian distributions, the users/items can represent more richer meanings. As depicted in Fig.1(a), Mary’s distribution overlaps her prefer books (e.g., Woman in Love and Global History) rather than horror items to avoid inaccurate results. While only using single domain information cannot better model users’ preference distribution, as shown in Fig.1(b), Mary’s local distribution with dashed eclipse does not overlap with the romantic movie Titanic that
she likes. Then we should adopt the global server to aggregate the user distributions for knowledge sharing in Fig.1(c). Meanwhile, different domains are always heterogeneous with diverse kinds of items. Hence, it could be rather difficult to directly apply the commonly-used federated learning methods for information gathering. What is more, most of the CDR models only concentrate on user-item interactions and neglect potential semantic relations. As shown in Fig.1(d)-(e), the user preference distributions are scattered and users with different tastes still have overlapped space. In conclusion, how to properly model and aggregate the user/item preference distribution is still a challenging problem.

To address the aforementioned issues, in this paper, we propose Federated Probabilistic Preference Distribution Modelling (FPPDM) for solving the PPMDR problem. FPPDM involves two main components, i.e., local domain modelling component and global server aggregation component where user-item rating interactions are stored in the local domain due to the demand for privacy protection. We further propose neural networks to capture user/item preference distribution via modelling the corresponding mean and covariance in the local domain modelling component. Then we aggregate these users’ preference distributions in the global server aggregation component and send them back to local domains. To further leverage the user similarity relationship, we propose a compactness co-clustering method in FPPDM++. The compactness co-clustering method can gather users based on their tastes or characteristics to provide more satisfactory results. We summarize our main contributions as follows: (1) We propose a novel federal learning framework, i.e., FPPDM, for solving the PPMDR problem and protecting the privacy of user-item rating information. FPPDM involves multiple local domains and a global server to model and aggregate user/item preference distributions. (2) To achieve better results, we further propose a compactness co-clustering method to leverage similar users’ information in FPPDM++. (3) Extensive empirical studies on Douban and Amazon datasets demonstrate that both FPPDM and FPPDM++ significantly improve the state-of-the-art models, especially under the PPMDR setting.

2 Related Work

Cross- and Multi-Domain Recommendation. The existing CDR models have two main types, i.e., single-target CDR models and dual-target CDR models [Zang et al., 2021b; Zhu et al., 2021; Zhang et al., 2022b; Yuan et al., 2019; Wang and et al., 2022; Sun and et al., 2022]. Single-target CDR models leverage the information from the relevant source domain with rich data to the sparse target domain. Conventional single-target CDR models adopt matrix factorization mechanism for collaborative filtering on overlapped users/items [Lian et al., 2017]. More recently, some single-target CDR models also adopt the deep neural networks to users/items preferences [Elkahky et al., 2015; Hu et al., 2018]. Latest PriCDR [Chen and et al., 2022] further considers the privacy protection when distilling the information from rich to sparse domains. Dual-target CDR models aims to simultaneously share and transfer information when both source and target domains are sparse. DOML [Li and Tuzhilin, 2021] first tried attempt to share dual knowledge transfer based on orthogonal mapping. Recently, researchers also utilized graph-based approaches [Zhao et al., 2019; Liu et al., 2020; Cao et al., 2022] for solving dual-target CDR problem. Zhao et.al. [Zhao et al., 2022] further enhanced the method with gate mechanism into multi-domain recommendation. While the above methods should distribute the local user-item ratings across different domains which could raise the concern on privacy protection. Hence they cannot be suitable for solving the PPMDR problem.

Federated Learning. With the increasing emphasis on protecting personal privacy, federated learning is widely used in diverse applications [Zhang et al., 2022a; Meihan et al., 2022]. FedAvg [McMahan et al., 2017] was first be proposed for protecting the clients’ privacy in distributed learning. To better satisfy the non-identical independent fashion, proximal regularization term [T Dinh et al., 2020] or primal-dual algorithm [Zhang et al., 2021; Acar et al., 2021] have been added in the local clients. Furthermore, shared representations have been exploited [Pillutla et al., 2022] to provide personal solutions. Meanwhile knowledge distill method [Lin et al., 2020] and prototype alignment [Tan and et al., 2021; Dong et al., 2022] have been proposed with the consideration of heterogeneous data distribution or network architecture. More recently, FedMF [Chai et al., 2020] successfully adopted the federated learning into the field of cross domain recommendation. Latest FedCDR [Meihan et al., 2022] also adopted federated learning into solving cold-start CDR problem. However, FedMF only adopted shallow model which cannot capture and depict the complex and complicate user-
item relations. While FedCDR separated the rating prediction stage and transfer stage which limits the model performance. As a comparison, in this paper, we extend the federated learning with PPMDR problem by adopting distribution aggregation method with compactness co-clustering strategy.

3 Methodology
First, we describe notations. We assume there are $K$ ($K \geq 2$) domains (clients) as $\{\mathbb{D}^{(1)}, \mathbb{D}^{(2)}, \ldots, \mathbb{D}^{(K)}\}$ where $\mathbb{D}^{(i)}$ denotes the $i$-th domain (client). We assume each domain share the same set of $N_U$ users. Meanwhile each domain has $N^{(i)}_V$ different types of items. Let $\mathbf{R}^{(i)} \in \mathbb{R}^{N_U \times N^{(i)}_V}$ be the observed rating matrices in $i$-th domain $\mathbb{D}^{(i)}$ and users in different domains are overlapped. To simplify the problem, in this paper, we assume both domains have no other auxiliary information. Meanwhile, the rating matrix $\mathbf{R}^{(i)}$ is the private data which cannot be directly shared in the public space.

Then, we introduce the overview of our proposed FPPDM framework, as is illustrated in Fig. 2. FPPDM mainly involves two components, i.e., local domain modelling component and global server aggregation component. The $i$-th local domain stores the user-item rating $\mathbf{R}^{(i)}$ in $i$-th domain $\mathbb{D}^{(i)}$. The local domain aims to model user/item probabilistic preference distribution based on the rating interactions. The global server aggregation component aims to gather the overlapped user information for knowledge sharing. To achieve more robust recommendation results, we further provide FPPDM-+ with compactness co-clustering method. This method tends to exploit and cluster users with similar tastes and characteristics.

3.1 The Framework of FPPDM
Local Domain Modelling Component
Firstly, we provide the details of the local domain module in FPPDM. For the $i$-th user and the $j$-th item in the $\mathbb{D}^{(i)}$ domain, we define their corresponding one-hot ID vectors as $\mathbf{X}_i^{U^{(i)}}$ and $\mathbf{X}_j^{V^{(i)}}$, respectively. We adopt a trainable lookup table to exploit the user/item one-hot ID embedding as $\mathbf{E}_i^{U^{(i)}} = \text{LookUp}(\mathbf{X}_i^{U^{(i)}})$ and $\mathbf{E}_j^{V^{(i)}} = \text{LookUp}(\mathbf{X}_j^{V^{(i)}})$ respectively. To better aggregate useful information among the user-item interactions, we further adopt the commonly-used graph neural network in modelling. Then we first build up the corresponding interaction $\mathbf{A}^{(i)}$ graph among users and items as $\mathbf{A}^{(i)} = \begin{bmatrix} 0 & \mathbf{R}^{(i)} \\ (\mathbf{R}^{(i)})^\top & 0 \end{bmatrix}$. Then we can conduct the graph convolution neural network for modelling user/item preference distribution:

$$
\begin{align}
[\mu^{U^{(i)}}, \mu^{V^{(i)}}] &= \text{GCN}(\cdots, \text{GCN}(\mathbf{E}^{U^{(i)}}, \mathbf{A}^{(i)} \mathbf{W}_\mu^{(i)}), \cdots), \\
[\sigma^{U^{(i)}}, \sigma^{V^{(i)}}] &= \text{GCN}(\cdots, \text{GCN}(\mathbf{E}^{U^{(i)}}, \mathbf{A}^{(i)} \mathbf{W}_\sigma^{(i)}), \cdots),
\end{align}
$$

(1)

where $\mathbf{E}^{(i)} = \{\mathbf{E}_i^{U^{(i)}}, \mathbf{E}_j^{V^{(i)}})$ denotes the sets of users and items in $\mathbb{D}^{(i)}$. The $\mathbf{W}_\mu^{(i)}$ and $\mathbf{W}_\sigma^{(i)}$ denote the trainable weights. GCN($\cdot$) denotes the graph convolution network operation which can be computed as:

$$
\text{GCN}(\mathbf{E}^{(i)}, \mathbf{A}^{(i)} \mathbf{W}^{(i)}) = (\tilde{\mathbf{D}}^{(i)})^{-\frac{1}{2}} \tilde{\mathbf{A}}^{(i)} (\tilde{\mathbf{D}}^{(i)})^{-\frac{1}{2}} \mathbf{E}^{(i)} \mathbf{W}^{(i)},
$$

(2)

where $\tilde{\mathbf{D}}^{(i)} = \text{diag}(\tilde{\mathbf{A}}^{(i)} \mathbf{1})$ denotes the degree matrix for the graph $\tilde{\mathbf{A}}^{(i)}$ and $\mathbf{A}^{(i)} = \mathbf{A}^{(i)} + \mathbf{I}$. Specifically, we adopt $\ell$-th layers of graph convolution network layers to achieve the users/items’ mean $\mu$ and covariance $\sigma^2$ of their embeddings distribution as follows:

$$
\mathbb{P}(\mathbf{U}^{(i)}) = \mathcal{N}(\mu^{U^{(i)}}, [\sigma^{U^{(i)}}, \sigma^{V^{(i)}}]^2),
$$

(3)

$\mathbb{P}(\mathbf{V}^{(i)}) = \mathcal{N}(\mu^{V^{(i)}}, [\sigma^{U^{(i)}}, \sigma^{V^{(i)}}]^2),$

where $\mathbb{P}(\mathbf{U}^{(i)})$ and $\mathbb{P}(\mathbf{V}^{(i)})$ denote the local user and item distributions respectively. Since using the single user or item embeddings cannot precisely depict user-item relationship, we adopt the Gaussian distribution to parameterize user and item distribution. Specifically, the Gaussian distribution can capture the complex and complicated user/item preference via measuring the covariance. After that we should train the model to fit the observed user-item ratings. To better achieve this goal, we propose the distribution-based metric learning loss as given below:

$$
\ell_R^{(i)} = - \sum_{(u_i^{(k)}, v_j^{(k)}) \in \mathcal{O}^{(k)}} \log \frac{e^{-D(\mathbb{P}(\mathbf{U}^{(i)}), \mathbb{P}(\mathbf{V}^{(i)}))}}{\sum_{(u_i^{(k)}, v_j^{(k)}) \in \mathcal{O}^{(k)}} e^{-D(\mathbb{P}(\mathbf{U}^{(i)}), \mathbb{P}(\mathbf{V}^{(i)}))}},
$$

(4)

where $\mathcal{O}^{(k)}$ denotes the positive user-item pairs, $\mathcal{V}^{(k)}_{neg.(i)}$ denotes the negative items for the $i$-th user. The $D(\cdot)$ denotes Wasserstein distance among different Gaussian distributions which can be calculated as:

$$
D(\mathbb{P}(\mathbf{U}^{(i)}), \mathbb{P}(\mathbf{V}^{(i)})) = ||\mu^{U^{(i)}} - \mu^{V^{(i)}}||_2^2 + ||\sigma^{U^{(i)}} - \sigma^{V^{(i)}}||_2^2.
$$

After adopting the metric-based rating prediction loss, we can pull the positive user-item pairs while push away the negative user-item pairs. Meanwhile, the local and global overlapped user distributions should be consistent for knowledge sharing. To this end, we propose a regularization term to reduce the distance among these corresponding local and global user distributions as follows:

$$
\ell_P^{(i)} = \sum_{i=1}^{N} ||\mu^{U^{(i)}} - \tilde{\mu}||_2^2 + ||\sigma^{U^{(i)}} - \tilde{\sigma}||_2^2,
$$

(5)

where $N$ denotes as the batchsize. $\mathbb{P}(\tilde{\mathbf{U}})$, $\mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2)$ denotes the $i$-th global user distribution. By combining the rating prediction loss $\ell_R^{(i)}$ with the regularization term $\ell_P^{(i)}$, we can obtain the total loss in local domain modelling component:

$$
\min \ell_{\text{FPPDM}}^{(i)} = \ell_R^{(i)} + \lambda_P \ell_P^{(i)},
$$

(6)

where $\lambda_P$ denotes as the balanced hyper parameter. After the training in local domains, we can obtain the local user distribution $\mathbb{P}(\mathbf{U})$ and send them to the global server.

Global Server Aggregation Component
Since the global server have received the local user distributions from multiple different domains, the global server should aggregate and update the global user distributions $\mathbb{P}(\tilde{\mathbf{U}})$. This process is important since the data sparsity problem will deteriorate the user-item modelling in local domain. As an example in Fig. 1(b), the local user distribution of Mary will not fully cover her interests since it still away from the romantic movie Titanic. Then we adopt the server aggregation
to obtain the global user preference distribution in Fig.1(c) and after that it will enhance the model performance in the local movie domain in Fig.1(b). To fulfill this task, we tend to exploit the new global user distributions \( \hat{P}(\hat{U}) \) which has the smaller Wasserstein distance among these local user distributions as \( \min_{P(\hat{U})} \sum_{i=1}^{N} \sum_{k=1}^{K} D(P(\hat{U}), P(U_i^{(k)})) \). By taking the differentiation w.r.t. on \( \hat{\mu}_i \) and \( \hat{\sigma}_i \), we can easily obtain the optimal results as:

\[
\hat{\mu}_i = \frac{\sum_{k=1}^{K} U_{i}^{(k)}}{K}, \quad \hat{\sigma}_i = \frac{\sum_{k=1}^{K} \sigma_{i}^{(k)}}{K}.
\]

After we achieve the aggregated global user distributions, we further send them back to the local domains. Meanwhile only mean and covariance of user preference distributions will be transmitted to the global server which makes it different from conventional federated learning methods (e.g., FedAvg [McMahan et al., 2017]). Note that it is difficult to reconstruct the raw user-item rating interactions from user preference distributions. Furthermore, one can even adopt encryption transportation method (e.g., homomorphic encryption [Gentry, 2009; Aono et al., 2017]) on mean and covariance to strengthen security level.

### 3.2 The Framework of FPPDM++

Although FPPDM can share and transfer useful knowledge via the overlapped users across domains, the user preference distributions will become scattered in the latent space as shown in Fig.1(d). Specifically, the purple and green dots represent the users with different types of preferences in Fig.1(d). Some users (e.g., purple #7 and green #2) with different preferences will still have overlapped distributions. The scattered latent distributions will hurdle the model performance by providing inaccurate results. The main reason behind is FPPDM cannot fully exploit and incorporate the semantic neighbors to enhance the model performance. That is, the users with similar tastes or preferences are their semantic neighbors who should be clustered. Meanwhile utilizing the co-clustering strategy can even filter out inheriting data noise to further reduce data sparsity problem [Lin et al., 2022b; Lin et al., 2022a]. However, previous methods mainly focus on the simple embedding situation and they cannot be directly applied for distribution clustering. To resolve this issue, we first propose compactness co-clustering method in FPPDM++ to enhance the model performance by extracting useful semantic information among these preference distributions.

Algorithm 1 FPPDM/FPPDM++

1. **Server executes:**
2. Initialize global user distribution as \( P(\hat{U}) \).
3. for \( \text{round} = 1 \) to \( T \) do
   4. for each domain \( k \) to \( K \) in parallel do
      5. Obtain \( \hat{P}(\hat{U}_i^{(k)}) \) \( \leftarrow \) LocalUpdate\((i, P(\hat{U}))\)
   6. end for
   7. Update \( \hat{P}(\hat{U}) \) via Eq.(7) and send to local domain.
   8. end for
   9. LocalUpdate\((i, P(\hat{U}))\):
      10. for \( \text{epoch} = 1 \) to \( r \) do
          11. Calculate the rating prediction loss in Eq.(4).
          12. Calculate the regularization loss in Eq.(5).
          13. if The model is FPPDM++ then
              14. Optimize the compactness similarity in Eq.(8).
              15. Optimize the assign indication in Eq.(10).
          17. end if
          18. Update the local model.
      19. end for

Then we introduce the details of our proposed compactness co-clustering method. To start with, we tend to figure out the similarity \( s_{ij}^{(k)} \) between the \( i \)-th and \( j \)-th users in the \( D^{(k)} \) domain. Meanwhile we suppose to cluster the users into \( M \) groups to achieve more compact user representations. Specifically, we present the compactness similarity optimization problem as follows:

\[
\min_{s_{ij}^{(k)}} \sum_{i,j=1}^{N} \left[ s_{ij}^{(k)} D(P(U_i^{(k)}), P(U_j^{(k)})) + \epsilon s_{ij}^{(k)} \log s_{ij}^{(k)} \right]
\]

\[
s.t. \sum_{j=1}^{N} s_{ij}^{(k)} = 1, s_{ij}^{(k)} \geq 0, \text{rank}(L^{(k)}) = N - M,
\]

where \( L^{(k)} = \text{diag}(S^{(k)}) - S^{(k)} \) denotes the Laplacian similarity matrix. \( \epsilon \) denotes the hyper parameter to balance the similarity matching and the entropy regularization term \( \epsilon s_{ij}^{(k)} \log s_{ij}^{(k)} \). Meanwhile the entropy regularization term is set to obtain the nonnegative and nontrivial solution [Bai and Liang, 2020; Nie et al., 2014]. Furthermore, we add the rank constraint, i.e., rank\((L^{(k)}) = N - M \), on Laplacian similarity matrix \( L^{(k)} \) to obtain more compact results.
by avoiding the situation that most users clustered in one group. However, it is difficult to directly optimize the rank constraint. Here, we make the approximation by calculating the minimal eigenvalues of $L^{(k)}$ to replace the rank constraint. In other words, $\text{rank}(L^{(k)}) = N - M$ is equivalent to $\sum_{m=1}^{M} \phi_m(L^{(k)})$ where $\phi_m(L^{(k)})$ denotes the $m$-th smallest eigenvalue of $L^{(k)}$. According to the Ky Fan’s theorem [Fan, 1949], $\phi_m(L^{(k)})$ can be calculated as:

$$\sum_{m=1}^{M} \phi_m(L^{(k)}) = \min_{F(k)} \text{Tr}((F(k)^{T} L^{(k)} F(k)),$$

where $F(k) \in \mathbb{R}^{N \times M}$ is the indicator matrix. We provide the Lagrange multipliers for the original problem as follows:

$$\min_{s(k), p(k)} f^{(k)} = \sum_{i,j=1}^{N} s_{ij}^{(k)} D(P(U_i^{(k)}), P(U_j^{(k)})) + \epsilon(s_{ij}^{(k)} \log s_{ij}^{(k)}) + \eta_{ij}^{(k)} ||f_i^{(k)} - f_j^{(k)}||_2^2 + \sum_{i=1}^{N} \gamma_i^{(k)} \left( \sum_{j=1}^{N} s_{ij}^{(k)} - 1 \right),$$

where $\eta$ is the hyper parameter and $\gamma_i^{(k)}$ is the multiplier. We further make simplification for rank constraint since $\sum_{i,j=1}^{N} s_{ij}^{(k)} ||f_i^{(k)} - f_j^{(k)}||_2^2 = 2\text{Tr}((F(k)^{T} L^{(k)} F(k))$.

We first fix $F(k)$ and update $s(k)$ as:

$$s_{ij}^{(k)} = \frac{\exp(-D(P(U_i^{(k)}), P(U_j^{(k)})) + \eta_{ij}^{(k)} ||f_i^{(k)} - f_j^{(k)}||_2^2)}{\sum_{l=1}^{N} \exp(-D(P(U_l^{(k)}), P(U_j^{(k)})) + \eta_{ij}^{(k)} ||f_i^{(k)} - f_j^{(k)}||_2^2).$$

Then we fix $s(k)$ and update $F(k)$. The optimal solution $F(k)$ can be obtained by the $M$ eigenvectors of $L^{(k)}$ corresponding to the $M$ smallest eigenvalues. We can alternatively update $s(k)$ and $F(k)$ to achieve the optimal solution.

Once we obtain the indicator matrix $F(k)$, we tend to figure out the relationship between each data and the corresponding clusters. To achieve this goal, we propose entropy-based K-Means method as follows:

$$\min_{\zeta^{(k)}: 1,1, Z^{(k)}} \sum_{i=1}^{N} \sum_{j=1}^{M} \zeta^{(k)}_{ij} ||f_i^{(k)} - z_j^{(k)}||_2^2 + \epsilon \zeta^{(k)}_{ij} \log \zeta^{(k)}_{ij},$$

where $Z^{(k)}$ denotes the cluster centers of assign indicator matrix $F(k)$. That problem can be solved via iteratively optimize $\zeta^{(k)}$ and $Z^{(k)}$. Due to space limits, please kindly refer to [Bai and et al., 2020; Liu and et al, 2023] for more details on the optimization process. After we obtain the optimal solution of $\zeta^{(k)}$, we can further calculate the results of the corresponding cluster distribution $P(C_j^{(k)}) = N(\mu_j^{(k)}, (\sigma_j^{(k)})^2)$ as:

$$\mu_j^{(k)} = \frac{\sum_{i=1}^{N} s_{ij}^{(k)} u_i^{(k)}}{\sum_{i=1}^{N} s_{ij}^{(k)}}, \quad \sigma_j^{(k)} = \frac{\sum_{i=1}^{N} s_{ij}^{(k)} \sigma_i^{(k)}}{\sum_{i=1}^{N} s_{ij}^{(k)}}.$$

After we obtain these results, we tend to narrow the distance between the local user distribution and these cluster distributions. Therefore, we further provide the compactness co-clustering loss as follows:

$$\ell_{C}^{(k)} = \sum_{i=1}^{N} \sum_{j=1}^{M} \zeta^{(k)}_{ij} \cdot D(P(U_i^{(k)}), P(C_j^{(k)})).$$

### Table 1: The statistics information of Douban and Amazon datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Users</th>
<th>Items</th>
<th>Ratings</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Douban Movie</td>
<td>800</td>
<td>154,886</td>
<td>93,074</td>
<td>0.075%</td>
</tr>
<tr>
<td>Douban Book</td>
<td>800</td>
<td>165,461</td>
<td>29,781</td>
<td>0.022%</td>
</tr>
<tr>
<td>Douban Music</td>
<td>800</td>
<td>166,447</td>
<td>30,487</td>
<td>0.023%</td>
</tr>
<tr>
<td>Amazon Phone</td>
<td>16,337</td>
<td>9,481</td>
<td>148,271</td>
<td>0.096%</td>
</tr>
<tr>
<td>Amazon Electronics</td>
<td>16,337</td>
<td>46,460</td>
<td>821,301</td>
<td>0.124%</td>
</tr>
<tr>
<td>Amazon Sport</td>
<td>7,857</td>
<td>12,655</td>
<td>163,291</td>
<td>0.164%</td>
</tr>
<tr>
<td>Amazon Cloth</td>
<td>7,857</td>
<td>17,943</td>
<td>187,880</td>
<td>0.133%</td>
</tr>
<tr>
<td>Amazon Game</td>
<td>1,730</td>
<td>12,319</td>
<td>25,036</td>
<td>0.117%</td>
</tr>
<tr>
<td>Amazon Video</td>
<td>1,730</td>
<td>8,751</td>
<td>16,091</td>
<td>0.106%</td>
</tr>
</tbody>
</table>

Using the compactness co-clustering, the users with similar tastes or preferences will have more compact representations as shown in Fig.1(e). Finally, we can combine the rating prediction loss $\ell^{(k)}_{R}$, local-global regularization loss $\ell^{(k)}_{P}$, and compactness co-clustering loss $\ell^{(k)}_{C}$ for the total loss of proposed FPPDM++:

$$\min \ell_{FPPDM++} = \ell^{(k)}_{R} + \lambda_{P} \ell^{(k)}_{P} + \lambda_{C} \ell^{(k)}_{C},$$

where $\lambda_{P}$ and $\lambda_{C}$ denote the balance hyper parameters. Note that the FPPDM and FPPDM++ have the same global server aggregation component during training process. The training algorithm details have been provided in Alg.1.

### 4 Experiments

#### 4.1 Experimental Setup

**Datasets.** We conduct experiments on two popularly used real-world datasets, i.e., Douban and Amazon. First, the Douban dataset [Zhu et al., 2019; Zhu and et al, 2021] has three domains, i.e., Book, Music, and Movie. Second, the Amazon dataset [Zhao et al., 2020; Ni et al., 2019] has six domains, i.e., Phone, Electronics (Elec), Cloth, Sport, Game, and Video. The detailed statistics of these datasets are given in Table 1. For each dataset, we binarize the ratings higher or equal to 4 as positive. We also filter the users and items with less than 5 interactions, following existing research [Zhu et al., 2019; Liu and et al, 2022]. We conduct three cross-domain recommendation tasks in Amazon when only two domains (clients) involve (K = 2), i.e., Phone and Elec, Cloth and Sport, and Game and Video. Meanwhile we conduct multi-domain recommendation on the whole Douban datasets (K = 3). We assume that users in different domains are overlapped in the same task, and there are two/three clients involves in Amazon/Douban tasks respectively.

**Baseline.** We compare our proposed FPPDM with the following state-of-the-art models. (1) NeuMF [He et al., 2017] first adopts deep neural network for collaborative filtering in single domain. (2) PMLAM [Ma et al, 2020] first adopts probabilistic user/item modelling with metric-based loss in single domain. (3) LightGCN [He et al., 2020] adopts the graph neural network to model user/item interactions in single domain. (4) FedMF [Chai et al., 2020] is the first attempt to adopt federal learning framework across domains. (5) FedGNN [Wu et al., 2021] further adopts the graph neural network into federated recommendation. (6) PriCDR [Chen and et al, 2022] is the single-target CDR model while sharing the information from the rich domain to the sparse domain.
ETL DOML cross domain model with information bottleneck theorem.

We set the hyper parameter $\lambda_D$ to 0.05 for local client modeling component in both FPPDM and FPPDM++. The number of graph encoder layer is set to 3 in each clients, following [He et al., 2020; Wang et al., 2019b]. Meanwhile set the hyper parameter $\lambda_C = 0.07$ and the number of clusters as $M = 15$ for compactness co-clustering loss in FPPDM++. Besides, we set the balance hyper parameter $\epsilon = 0.05$ and $\eta = 0.1$.

We conduct training for FedCDR, FPPDM and FPPDM++ with 5 local epochs per round until converge. We choose Adam [Kingma and Ba, 2014] as optimizer, and adopt HR and NDCG [Wang et al., 2019a] as the evaluation metrics. For all the experiments, we perform five random experiments and report the average results. We report the results measured by the commonly used metrics as Top@5 and Top@10 in Douban and Amazon datasets, respectively.

### 4.2 Recommendation Performance

The comparison results on Douban and Amazon datasets are shown in Table 2. Note that conventional CDR models (e.g., DARec, DOML, and CDRIB) cannot directly adjust to the multi-domain recommendation in Douban. Therefore, we set Douban Movie as the source domain and Douban Book, Douban Music as the target domain for dealing with these CDRIB [Cao et al., 2022] is the state-of-the-art graph-based cross-domain model with information bottleneck theorem. (9) MSCDR [Zhao et al., 2022] is the state-of-the-art graph-based multi domain model with attention mechanism. We adopt the same user-item ratings without other auxiliary information for our propose methods and these baselines.

### Implemented details

We set batch size $N = 256$ and embedding dimension as $D = 128$ across different domains. We set the hyper parameter $\lambda_D = 0.05$ for local client modeling component in both FPPDM and FPPDM++. The number of graph encoder layer is set to 3 in each clients, following [He et al., 2020; Wang et al., 2019b]. Meanwhile set the hyper parameter $\lambda_C = 0.07$ and the number of clusters as $M = 15$ for compactness co-clustering loss in FPPDM++. Besides, we set the balance hyper parameter $\epsilon = 0.05$ and $\eta = 0.1$.

We conduct training for FedCDR, FPPDM and FPPDM++ with 5 local epochs per round until converge. We choose Adam [Kingma and Ba, 2014] as optimizer, and adopt HR and NDCG [Wang et al., 2019a] as the evaluation metrics. For all the experiments, we perform five random experiments and report the average results. We report the results measured by the commonly used metrics as Top@5 and Top@10 in Douban and Amazon datasets, respectively.

### Table 2: Experimental results (%) on Douban and Amazon datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>HR@5</th>
<th>NDCG@5</th>
<th>HR@10</th>
<th>NDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Amazon) Book</td>
<td>NEOMF</td>
<td>25.36</td>
<td>16.42</td>
<td>45.52</td>
<td>29.23</td>
</tr>
<tr>
<td></td>
<td>PMLAM</td>
<td>25.80</td>
<td>16.92</td>
<td>46.84</td>
<td>29.57</td>
</tr>
<tr>
<td></td>
<td>LightGCN</td>
<td>26.47</td>
<td>17.51</td>
<td>46.69</td>
<td>29.68</td>
</tr>
<tr>
<td>(Douban) Book</td>
<td>NEOMF</td>
<td>26.18</td>
<td>17.10</td>
<td>45.84</td>
<td>29.24</td>
</tr>
<tr>
<td></td>
<td>PMLAM</td>
<td>26.67</td>
<td>17.61</td>
<td>46.37</td>
<td>29.75</td>
</tr>
<tr>
<td></td>
<td>LightGCN</td>
<td>27.29</td>
<td>18.23</td>
<td>47.18</td>
<td>29.37</td>
</tr>
</tbody>
</table>

### Table 3: Method extension results (%) on Douban datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>HR@5</th>
<th>NDCG@5</th>
<th>HR@10</th>
<th>NDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Amazon) Book</td>
<td>NEOMF</td>
<td>24.84</td>
<td>15.40</td>
<td>41.63</td>
<td>26.03</td>
</tr>
<tr>
<td></td>
<td>PMLAM</td>
<td>25.79</td>
<td>15.89</td>
<td>42.46</td>
<td>26.51</td>
</tr>
<tr>
<td></td>
<td>LightGCN</td>
<td>26.03</td>
<td>16.22</td>
<td>42.97</td>
<td>26.82</td>
</tr>
</tbody>
</table>

### Table 4: The results (%) of tuning $M$ on Douban datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>HR@5</th>
<th>NDCG@5</th>
<th>HR@10</th>
<th>NDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Amazon) Book</td>
<td>NEOMF</td>
<td>25.36</td>
<td>16.42</td>
<td>45.52</td>
<td>29.23</td>
</tr>
<tr>
<td></td>
<td>PMLAM</td>
<td>25.80</td>
<td>16.92</td>
<td>46.84</td>
<td>29.57</td>
</tr>
<tr>
<td></td>
<td>LightGCN</td>
<td>26.47</td>
<td>17.51</td>
<td>46.69</td>
<td>29.68</td>
</tr>
<tr>
<td>(Douban) Book</td>
<td>NEOMF</td>
<td>26.18</td>
<td>17.10</td>
<td>45.84</td>
<td>29.24</td>
</tr>
<tr>
<td></td>
<td>PMLAM</td>
<td>26.67</td>
<td>17.61</td>
<td>46.37</td>
<td>29.75</td>
</tr>
<tr>
<td></td>
<td>LightGCN</td>
<td>27.29</td>
<td>18.23</td>
<td>47.18</td>
<td>29.37</td>
</tr>
</tbody>
</table>

---

Li and Tuzhilin, 2021 adopts the dual deep transfer module with orthogonal constraints across domains. (8) ETL [Chen et al., 2020] utilizes equivalent transformation with autoencoder framework for cross-domain modelling. (9) CDRIB [Cao et al., 2022] is the state-of-the-art graph-based cross-domain model with information bottleneck theorem. (10) MSCDR [Zhao et al., 2022] is the state-of-the-art graph-based multi domain model with attention mechanism. We adopt the same user-item ratings without other auxiliary information for our propose methods and these baselines.
by comparing PMLAM with NeuMF. However, single-domain methods cannot fully utilize the knowledge across domains which make them hard to tackle the sparsity problem. (2) Combining multiple domains information can enhance the model performance by comparing NeuMF with DOML. Meanwhile, recent models (e.g., CDRIB) which adopt graph-based framework even achieve much more promising results than single domain graph-based model (e.g., Light-GCN). (3) The federal learning framework (e.g., FedMF) can even exceed the performance than single domain methods (e.g., NeuMF) by combing knowledge and protecting privacy. Nevertheless, the non-deep-learning-based FedMF cannot handle the complex and complicated user-item relationship. Meanwhile, single-target based PriCDR simply transferred knowledge from rich to the sparse domain. Thus it cannot better handle the situation when different domains are both sparse. Latest FedCDR model also separates the training and mapping process limits the model performance. (4) Although latent models (e.g., MetaDPA and MSCDR) reach better performance, they still only extract user/item embeddings which limit their model potentials. These deterministic manner cannot capture and depict complicated user-item relationships. (5) Our proposed FPPDM achieves much satisfied results which indicates the efficacy of FPPDM by integrating preference distribution across different domains. Furthermore, FPPDM++ with compactness co-clustering method can gather users with similar characteristics which improve the model performance. Hence FPPDM++ even achieve much better results than proposed FPPDM.

Moreover, combing distribution modelling and compactness clustering can greatest improve the model performance.

4.3 Analysis

Ablation. We compare FPPDM/FPPDM++ with its several variants, including FPPDM++(B) and FPPDM++(P) to study how does each component of FPPDM++ contribute to the final performance. FPPDM++(B) and FPPDM++(P) both adopt the deterministic training manners without estimating the user covariance. While FPPDM++(B) excludes compactness co-clustering module and set $\lambda_C = 0$. We conduct the FPPDM++(B) and FPPDM++(P) on Amazon Phone and Elec and report the results (HR and NDCG) in Fig. 3. By comparing the FPPDM and FPPDM++(B) we can conclude that modelling user preference distribution can better capture diverse and complex relationships. However, they cannot gather similar users in the latent space limits the models’ accuracy. We can further observe that FPPDM++(P) exceeds the FPPDM++(B) which indicates that clustering users similar tastes or characteristics will boost the model potentials.

Method Extension. We further analyse the general extension of our proposed federated framework. To validate this, we apply our proposed FPPDM and FPPDM++ into the MLP-based autoencoder model ETL as ETL-F and ETL-FC on conducting the experiments on Douban dataset respectively. We report the results HR and NDCG on Table 3. From that we can conclude our proposed federated framework is model-agnostic. That is it can be applied in both graph-based or MLP-based recommendation approaches. Meanwhile utilizing the federated learning strategy by safely sharing knowledge can even enhance the model performance.

Effect of hyper-parameters. We finally study the effects of hyper-parameters on model performance. We first conduct experiments to study the effects of $\lambda_P$ by varying them in $\{0.01, 0.02, 0.05, 0.07, 0.1, 0.5\}$ for FPPDM++ on Amazon Cloth and Sport and report the results in Fig.4. It is difficult to apply local-global regularization alignment for knowledge sharing on each clients when $\lambda_P$ is too small. While when $\lambda_P$ is too large, it will slightly hurdle the rating prediction modelling. Therefore we set $\lambda_P = 0.05$ empirically. We further vary $\lambda_C$ in $\{0.01, 0.02, 0.05, 0.07, 0.1, 0.5\}$ and report the results in Fig.4. Similarly, the bell-shaped curve of $\lambda_C$ indicated that too large or too small value of $\lambda_C$ will not suitable in training the model and we set $\lambda_C = 0.07$. Finally, we conduct the experiments on varying the number of cluster $M$ in $\{5, 10, 15, 20\}$ and report the results on Table 4. When $M$ is smaller, it cannot better depict the users with similar tastes. By comparing $M = 20$ and $M = 15$, the performance improvement is rather marginal by consuming more time and space. Hence we set $M = 15$ empirically.

5 Conclusion

In this paper, we investigate the Privacy-Preserving Multi-Domain Recommendation (PPMDR) problem. To tackle this problem, we first propose Probabilistic Preference Distribution Modelling (FPPDM), which includes the local domain modelling component and global server aggregation component. FPPDM can model and share user/item preference distribution across different domains with federated learning strategy. To better exploit useful semantic information by clustering users with similar characteristics, we further propose FPPDM++ with compactness co-clustering method. We also conduct extensive experiments to demonstrate the superior performance of our proposed FPPDM models.
Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No.62172362) and CCF-AFSG Research Fund.

References


[Li and Tuzhilin, 2021] Pan Li and Alexander Tuzhilin. Dual metric learning for effective and efficient cross-domain recommendations. IEEE Transactions on Knowledge and Data Engineering, 2021.


[Liu et al., 2020] Meng Liu, Jianjun Li, Guohui Li, and Peng Pan. Cross domain recommendation via bi-directional


[Zhang et al., 2022b] Xinwei Zhang, Chanyu Li, Ivor W Tsang, Hui Xu, Lixin Duan, Hongzhi Yin, Wen Li, and Jie Shao. Diverse preference augmentation with multiple domains for cold-start recommendations. ICDE, 2022.


