Offline Recommender Learning Meets Unsupervised Domain Adaptation

Yuta Saito
Tokyo Institute of Technology
saito.y.bj@m.titech.ac.jp

Abstract

It is critical to eliminate selection bias of the rating feedback to construct a well-performing recommender offline. Currently, a promising solution to the challenge is the propensity weighting approach that models the missing mechanism of rating feedback. However, the performance of existing propensity-based algorithms can be significantly affected by the propensity estimation bias. To alleviate the problem, we formulate the missing-not-at-random recommendation as the unsupervised domain adaptation problem and drive the propensity-independent generalization error bound. We further propose a corresponding algorithm that minimizes the bound via adversarial learning. Our proposed theoretical framework and algorithm do not depend on the propensity score and can obtain a well-performing rating predictor without the true propensity information. Empirical evaluation using benchmark real-world datasets demonstrates the effectiveness and real-world applicability of the proposed approach.

1 Introduction

The main objective of recommender systems is to obtain a well-performing rating predictor from sparse observed rating feedback. During the process, an important challenge is that most of the missing mechanism of the real-world dataset is missing-not-at-random (MNAR). The MNAR missing mechanism is created owing, mainly, to two factors. The first is the past recommendation policy. If we relied on a policy recommending popular items with high probability in the past, then the observed ratings under that policy include more data of popular items [6, 36]. The other is users' self-selection. For example, users tend to rate items for which they exhibit positive preferences, and the ratings for negative preferences are more likely to be missing [22, 29].

The MNAR problem makes it difficult to learn rating predictors from observable data because it is widely recognized that naive methods typically lead to sub-optimal and biased recommendations under the MNAR settings [29, 30, 34]. One of the most established solutions to the problem is the propensity-based approach. It defines the probability of each instance being observed as the propensity score and obtains an unbiased estimator for the true metric of interest through weighting each data by the inverse of its propensity [20, 29, 35]. Generally, unbiasedness of the propensity-based methods is desirable, but this property is ensured only when the true propensities are available. It is widely known that the performance of propensity-based algorithms is highly susceptible to propensity estimation methods [2, 28]. However, in real-world recommender systems, the true propensities are mostly unknown, and this leads to severe bias in the estimation of the loss function of interest.

To solve the limitation of the existing propensity-based methods, in this study, we establish a new theory of MNAR recommendation inspired by the theoretical framework of unsupervised domain adaptation. Similar to the causal inference, unsupervised domain adaptation addresses the problem settings in which the feature distributions between the training and test sets are different. Furthermore, methods of unsupervised domain adaptation generally utilize the distance metrics that measure dissimilarity between probability distributions and do not depend on propensity weighting techniques [7, 8, 26, 27]. Thus, approaches for unsupervised domain adaptation is considered to be useful for addressing the issue caused by the propensity estimation bias. However, the connection between the MNAR recommendation and unsupervised domain adaptation has not yet been thoroughly investigated.

To bridge the two potentially related fields, we first define a discrepancy metric to measure the distance between the two missing mechanisms inspired by domain discrepancy measures for unsupervised domain adaptation [4, 5]. Subsequently, we derive a generalization error bound based on the discrepancy between the ideal missing-completely-at-random (MCAR) and common MNAR missing mechanisms. Our theoretical bound is independent of the propensity score, thus the bias problem related to the propensity scoring is eliminated. Furthermore, we propose an algorithm called *Domain Adversarial Matrix Factorization (DAMF)*. The proposed algorithm simultaneously minimizes the naive loss on the MNAR feedback and the discrepancy measure in an adversarial manner. Finally, we conduct an experiment on standard real-world datasets to empirically demonstrate the effectiveness of the proposed approach under realistic situations where true propensities are inaccessible.

The contributions of the study can be summarized as follows.

- We construct a new theoretical approach to the problem of MNAR recommendation based on the theoretical bound of unsupervised domain adaptation. Different from the previous propensity-based unbiased estimation approach, our proposed approach does not depend on the propensity score.
- We propose *Domain Adversarial Matrix Factorization* that eliminates the bias of a recommender by introducing domain adversarial learning to the matrix factorization model.
- We conduct comprehensive experiments on standard real-world datasets. Specifically, we demonstrate that the existing propensity-based approaches are susceptible to the choice of propensity estimators. Furthermore, our proposed method outperforms the baseline methods with respect to the rating prediction accuracy when the true propensity score is unknown.

2 Preliminaries

In this section, we introduce the notations and formulation of the MNAR recommendation with explicit feedback. Subsequently, we describe previous estimators and their limitations.

2.1 Notation and Formulation

In this study, $\mathcal{U}$ is a set of users ($|\mathcal{U}| = m$), and $\mathcal{I}$ is a set of items ($|\mathcal{I}| = n$). We also denote the set of all users and item pairs as $\mathcal{D} = \mathcal{U} \times \mathcal{I}$. Let $\boldsymbol{R} \in \mathbb{R}^{m \times n}$ be a true rating matrix, where each entry $R_{u,i}$ represents the true rating of user u to item i.

The objective of this study is to develop an algorithm to obtain a better predicted rating matrix $\widehat{\boldsymbol{R}}$, where each entry $\widehat{R}_{u,i}$ denotes the predicted rating for (u, i). To achieve the objective, we formally define the ideal loss function that an optimal algorithm should minimize as follows:

$$\mathcal{L}_{\text{ideal}}^{\ell}\left(\widehat{\boldsymbol{R}}\right) = \frac{1}{|\mathcal{D}|} \sum_{(u,i)\in\mathcal{D}} \ell\left(R_{u,i}, \widehat{R}_{u,i}\right) \tag{1}$$

where $\ell(\cdot, \cdot) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is an arbitrary loss function. For example, when $\ell(x, y) = |x - y|$, Eq. (1) is called the mean-absolute-error (MAE). Conversely, when $\ell(x, y) = (x - y)^2$, it is termed as the mean-squared-error (MSE).

In real-world recommender systems, it is impossible to calculate the ideal loss function because most of the rating data are missing. To precisely formulate this missing mechanism, we utilize two other matrices. The first corresponds to the **propensity matrix** denoted as $\boldsymbol{P} \in (0, 1)^{m \times n}$. Each entry of the matrix $P_{u,i}$ is the propensity score of (u, i) that represents the probability of the feedback being observed. Next, let $\boldsymbol{O} \in \{0, 1\}^{m \times n}$ be an observation matrix where each entry $O_{u,i} \in \{0, 1\}$ is a Bernoulli random variable with its expectation $\mathbb{E}[O_{u,i}] = P_{u,i}$. If $O_{u,i} = 1$, then the rating of the pair is observed; otherwise, it is unobserved. Throughout the study, we assume $M = \sum_{(u,i) \in \mathcal{D}} O_{u,i}$ for all the observation matrices.

Under the formulation, we aim to construct an effective estimator for the ideal loss function that can be estimated by using only a set of observable feedback, which is critical in developing an effective recommendation algorithm.

2.2 Naive Estimator

Given a feedback data $\boldsymbol{O}$, the most basic estimator for the ideal loss is the naive estimator, defined as follows:

$$\widehat{\mathcal{L}}^{\ell}_{\text{naive}} \left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{O} \right) = \frac{1}{M} \sum_{(u,i) \in \mathcal{D}} O_{u,i} \cdot \ell \left(R_{u,i}, \widehat{R}_{u,i} \right) \tag{2}$$

The naive estimator is the averaged loss values over the observed rating feedback. The estimator is valid when the missing mechanism of the rating data is missing-completely-at-random (MCAR) because the estimator is unbiased against the ideal loss function under the MCAR settings [29, 30].

However, several previous studies indicated that the simple naive estimator actually exhibits a bias under the general MNAR settings. Thus, it is undesirable to learn a recommendation algorithm, and one should rely on an estimator that addresses the bias as an alternative to using the naive estimator [29, 30].

2.3 Inverse Propensity Score Estimator

To improve the naive estimator, several previous studies applied the IPS estimation to the recommendation settings [20, 29]. In the context of causal inference, the propensity scoring estimator is widely used to estimate the causal effects of treatments from observational data [13, 24, 25]. It is possible to derive an unbiased estimator for the loss function of interest with the true propensity score as follows:

$$\widehat{\mathcal{L}}^{\ell}_{\text{IPS}} \left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{O} \right) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} O_{u,i} \cdot \frac{\ell \left(R_{u,i}, \widehat{R}_{u,i} \right)}{P_{u,i}} \tag{3}$$

The estimator is unbiased against the ideal loss and thus, is considered to be more desirable than the naive estimator in terms of bias. However, the unbiasedness is valid only when the true propensity score is available. The IPS estimator can have a bias with an inaccurate propensity estimator (see Lemma 5.1 of [29]). The bias problem of the IPS estimator typically occurs in most real-world recommender systems. This is because the missing mechanism of the rating feedback can depend on user self-selection, which is uncontrollable by the analysts. It is challenging to accurately estimate the missing probability of each instance [22, 29, 35].

Specifically, most previous studies estimate the propensity score for the propensity-based matrix factorization model by using some amount of MCAR test data [29, 34]. However, this type of propensity estimation is infeasible owing to the costly annotation process [9]. Therefore, in the next section, we explore the theory and algorithm that are independent of the propensity score aiming to alleviate the problem of propensity estimation bias. Additionally, we investigate the effect of using different propensity estimators on the performance of the propensity-based matrix factorization method in the experimental part.

3 Proposed Method

In this section, we first derive the generalization error bound of the ideal loss function based on the **discrepancy measure** between two different propensity matrices. Our bound is propensity-independent. Thus, the problem related to the propensity estimation is eliminated in the bound. Subsequently, we propose *Domain Adversarial Matrix Factorization (DAMF)*, which minimizes the theoretical upper bound via the adversarial learning procedure. The optimization of the proposed algorithm is independent of the propensity score. Hence, the advantage of the proposed method is emphasized in situations with unknown propensities. Note that all the proofs in this section are given in the supplementary materials.

3.1 Theoretical Bound

First, we define the discrepancy measure for the recommendation settings.

Definition 1. ($\mathcal{H}\Delta\mathcal{H}$-divergence for recommendation) Let $\mathcal{H}$ be a class of predicted rating matrices and let $\ell(\cdot,\cdot):\mathbb{R}\times\mathbb{R}\rightarrow\mathbb{R}_{\geq 0}$ be a loss function. Then, the $\mathcal{H}\Delta\mathcal{H}$-divergence between the two propensity matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$ is defined as follows:

$$d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{P},\boldsymbol{P}'\right)=\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\mid\boldsymbol{P}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\mid\boldsymbol{P}'\right)\right| \tag{4}$$

where

$$\begin{aligned}\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\mid\boldsymbol{P}\right)&=\mathbb{E}_{\boldsymbol{O}\sim\boldsymbol{P}}\left[\widehat{\mathcal{L}}^{\ell}_{\text{naive}}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\mid\boldsymbol{O}\right)\right]\\&=\frac{1}{M}\sum_{(u,i)\in\mathcal{D}}P_{u,i}\cdot\ell\left(\widehat{R}_{u,i},\widehat{R}'_{u,i}\right)\end{aligned}$$

Note that $\mathcal{H}\Delta\mathcal{H}$-divergence for recommendation is independent of the true rating matrices. Therefore, we can calculate this divergence for any given pair of propensity matrices without the true rating information.

However, in reality, the true propensity matrices ($\boldsymbol{P}$ and $\boldsymbol{P}'$) are unobserved. Thus, it is necessary to estimate the divergence using their realizations ($\boldsymbol{O}$ and $\boldsymbol{O}'$). The following lemma shows the deviation bound of $\mathcal{H}\Delta\mathcal{H}$-divergence.

Lemma 1. Any pair of propensity matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$ and their realizations $\boldsymbol{O}$ and $\boldsymbol{O}'$ are given. The loss function ℓ is bounded above by a positive constant Δ. Then, for any $\delta\in(0,1)$, the following inequality holds with a probability of at least $1-\delta$

$$\left|d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{P},\boldsymbol{P}'\right)-d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{O},\boldsymbol{O}'\right)\right|\leq\frac{\Delta}{M}\sqrt{2|\mathcal{D}|\log\frac{4|\mathcal{H}|^{2}}{\delta}} \tag{5}$$

Subsequently, we state the generalization error bound based on an ideal MCAR observation.

Lemma 2. (Generalization Error Bound under MCAR observation.) An MCAR-observation matrix $\boldsymbol{O}_{\text{MCAR}} \sim \boldsymbol{P}_{\text{MCAR}}$ where

$$(\boldsymbol{P}_{\text{MCAR}})_{u,i} = \mathbb{E}\left[O_{u,i}\right] = \frac{M}{|\mathcal{D}|}, \ \forall (u,i) \in \mathcal{D}$$

and any finite hypothesis space of predictions $\mathcal{H} = \{\widehat{\boldsymbol{R}}_1, \ldots, \widehat{\boldsymbol{R}}_{|\mathcal{H}|}\}$ are given. The loss function ℓ is bounded above by a positive constant Δ. Subsequently, for any hypothesis $\widehat{\boldsymbol{R}} \in \mathcal{H}$ and for any $\delta \in (0,1)$, the following inequality holds with a probability of at least $1-\delta$:

$$\mathcal{L}^{\ell}_{\text{ideal}}\left(\widehat{\boldsymbol{R}}\right) \leq \widehat{\mathcal{L}}^{\ell}_{\text{naive}}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{O}_{\text{MCAR}}\right) + \frac{\Delta}{M}\sqrt{\frac{|\mathcal{D}|}{2}\log\frac{2|\mathcal{H}|}{\delta}} \tag{6}$$

The next lemma relates the losses under two different propensity matrices.

Lemma 3. We assume that the loss function ℓ obeys the triangle inequality. Then, for any given predicted rating matrices $\widehat{\boldsymbol{R}} \in \mathcal{H}$ and two propensity matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$, the following inequality holds

$$\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}\right) \leq \mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}'\right) + d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{P}, \boldsymbol{P}'\right) + \lambda \tag{7}$$

where

$$\lambda = \min_{\widehat{\boldsymbol{R}} \in \mathcal{H}} \mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}\right) + \mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}'\right)$$

Finally, we use the $\mathcal{H}\Delta\mathcal{H}$-divergence for recommendation and derive the propensity-independent generalization error bound of the ideal loss function.

Theorem 1. (propensity-independent Generalization Error Bound) Two observation matrices $\boldsymbol{O}_{\text{MCAR}} \sim \boldsymbol{P}_{\text{MCAR}}$ and $\boldsymbol{O}_{\text{MNAR}} \sim \boldsymbol{P}_{\text{MNAR}}$ with MCAR and MNAR missing mechanisms, respectively, and any finite hypothesis space of predictions $\mathcal{H} = \{\widehat{\boldsymbol{R}}_1, \ldots, \widehat{\boldsymbol{R}}_{|\mathcal{H}|}\}$ are given. The loss function ℓ is bounded above by a positive constant Δ. Then, for any hypothesis $\widehat{\boldsymbol{R}} \in \mathcal{H}$ and for any $\delta \in (0,1)$, the following inequality holds with a probability of at least $1-\delta$

$$\begin{aligned}
&\mathcal{L}^{\ell}_{\text{ideal}}\left(\widehat{\boldsymbol{R}}\right) \\
&\leq \widehat{\mathcal{L}}^{\ell}_{\text{naive}}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{O}_{\text{MNAR}}\right) + d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{O}_{\text{MCAR}}, \boldsymbol{O}_{\text{MNAR}}\right) \\
&+ \frac{\Delta}{M}\left(\sqrt{\frac{|\mathcal{D}|}{2}\log\frac{4|\mathcal{H}|}{\delta}} + \sqrt{2|\mathcal{D}|\log\frac{8|\mathcal{H}|^2}{\delta}}\right) + \lambda
\end{aligned} \tag{8}$$

Following the related works on de-biasing recommendation algorithms with explicit feedback [29, 34], We consider only the case of using finite hypothesis space. However, as stated in [34], our results can be extended to the case of infinite hypothesis, for instance, using model complexity measures in [3, 23].

As previously explained, the bound derived in Theorem 4 is independent of the propensity score, and problems relating to the propensity score estimation are expected to be solved.

3.2 Algorithm

Here, we describe the detailed algorithm of the proposed DAMF. Inspired by Theorem 1, we consider minimizing the following objective:

$$\min_{\widehat{\boldsymbol{R}} \in \mathcal{H}} \underbrace{\widehat{\mathcal{L}}^{\ell}_{\text{naive}}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{O}_{\text{MNAR}}\right)}_{\text{loss on MNAR feedback}} + \beta \underbrace{d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{O}_{\text{MCAR}}, \boldsymbol{O}_{\text{MNAR}}\right)}_{\text{disc between MCAR and MNAR}}$$

where β denotes the trade-off hyperparameter between the naive loss on the MNAR feedback and the discrepancy between the MCAR and MNAR observation mechanisms. The objective consists of the two controllable terms of the theoretical bound in Eq. (8). Both terms are independent of the propensity score and it is unnecessary to estimate the propensity to optimize this objective. It is possible to easily perform the minimization of the first term (loss on MNAR feedback). Conversely, minimization of the second term (discrepancy between MCAR and MNAR) is difficult because optimization over the pair of hypotheses is required.

Therefore, in this work, we introduce a discriminator to classify item latent factors into two classes, rare and popular, with the aim to derive item latent factors, such that item popularity bias is eliminated. We adopt this approach as a practical solution because item popularity bias is the most problematic type of bias in recommender systems [36] and a similar optimization approach has shown promising results in the neural word embedding literature [10].

We describe the proposed algorithm below. First, we denote the user and item latent factors as $\boldsymbol{U} \in \mathbb{R}^{m \times d}$, $\boldsymbol{V} \in \mathbb{R}^{n \times d}$, and rating predictions are completed via the following dot product.

$$\widehat{R}\left(\boldsymbol{U}_u, \boldsymbol{V}_i\right) = \boldsymbol{U}_u \boldsymbol{V}_i^{\top}$$

The loss function to derive the parameters is as follows:

$$\widehat{\mathcal{L}}^{\ell}_{\text{pred}}\left(\widehat{\boldsymbol{R}}\right) = \frac{1}{M} \sum_{(u,i) \in \mathcal{O}} \ell\left(R_{u,i}, \widehat{R}\left(\boldsymbol{U}_u, \boldsymbol{V}_i\right)\right)$$

where $\mathcal{O} = \{(u, i, R_{u,i}) : (u, i) \in \mathcal{D}, O_{u,i} = 1\}$ denotes a dataset with observed ratings. Furthermore, predictions for the item popularity are completed via the following linear transformation:

$$f\left(\boldsymbol{V}_i; \theta_{\text{dom}}\right) = \sigma\left(\boldsymbol{W}_d \boldsymbol{V}_i^{\top} + b_d\right)$$

where $\theta_{\text{dom}} = (\boldsymbol{W}_d, b_d) \in \mathbb{R}^d \times \mathbb{R}$ denotes a vector-scalar parameter pair and $\sigma(\cdot)$ denotes the sigmoid function. The outputs are confidence scores that represent how rare each item is. The loss to derive these parameters is represented as the following binary cross entropy form.

$$\begin{aligned}\widehat{\mathcal{L}}_{\text{dom}}\left(f; \theta_{\text{dom}}\right) = &\frac{1}{|\mathcal{O}_{\text{rare}}|} \sum_{(u,i) \in \mathcal{O}_{\text{rare}}} \log\left(f\left(\boldsymbol{V}_i; \theta_{\text{dom}}\right)\right) \\ &+ \frac{1}{|\mathcal{O}_{\text{pop}}|} \sum_{(u,i) \in \mathcal{O}_{\text{pop}}} \log\left(1 - f\left(\boldsymbol{V}_i; \theta_{\text{dom}}\right)\right)\end{aligned}$$

where $\mathcal{O}_{\text{rare}} = \{(u, i) : (u, i) \in \mathcal{D}, i \in \mathcal{I}_{\text{rare}}\}$ denotes the set of all users and **rare** items ($\mathcal{I}_{\text{rare}} \subset \mathcal{I}$ is a set of rare items), in contrast, $\mathcal{O}_{\text{pop}} = \{(u, i) : (u, i) \in \mathcal{D}, i \in \mathcal{I}_{\text{pop}}\}$ denotes the set of all users and **popular** items ($\mathcal{I}_{\text{pop}} \subset \mathcal{I}$ is a set of popular items).

Algorithm 1 Domain Adversarial Matrix Factorization (DAMF)

Input: observed feedback data $\mathcal{O}$; sets of users and rare or popular items $\mathcal{O}_{\text{rare}}, \mathcal{O}_{\text{pop}}$; learning_rate η; trade-off hyperparameter β; *number of steps*.

Output: user-item latent factors $\boldsymbol{U}, \boldsymbol{V}$.

```
repeat
    Sample mini-batch from O_MNAR
    Update U and V by gradient descent according to Eq. (11) with fixed θ_dom
    for n = 1, ... number of steps do
        Update θ_dom by gradient ascent according to Eq. (10) with fixed U and V
    end for
until convergence;
return U, V
```

We follow the framework of domain adversarial training [7, 8, 10], and the rating predictor and the popularity discriminator are trained in a minimax manner as follows:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \max_{\theta_{\text{dom}}} \widehat{\mathcal{L}}^{\ell}_{\text{pred}} \left(\widehat{\boldsymbol{R}}\right) - \beta \widehat{\mathcal{L}}_{\text{dom}} \left(f; \theta_{\text{dom}}\right) \tag{9}$$

where $\beta > 0$ denotes the trade-off hyperparameter between the prediction and domain loss. Given fixed latent factors $\boldsymbol{U}$ and $\boldsymbol{V}$, the optimization of the discriminator is as follows:

$$\max_{\theta_{\text{dom}}} -\beta \widehat{\mathcal{L}}_{\text{dom}} \left(f; \theta_{\text{dom}}\right) \tag{10}$$

Subsequently, given fixed parameters θ_{dom}, the optimization of $\boldsymbol{U}$ and $\boldsymbol{V}$ is as follows:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \widehat{\mathcal{L}}^{\ell}_{\text{pred}} \left(\widehat{\boldsymbol{R}}\right) - \beta \widehat{\mathcal{L}}_{\text{dom}} \left(f; \theta_{\text{dom}}\right) \tag{11}$$

We implement the proposed algorithm by TensorFlow and optimize $\boldsymbol{U}, \boldsymbol{V}$, and θ_{dom} iteratively using the Adam optimizer [16]. The detailed training procedure of DAMF is described in Algorithm 1.

4 Experimental Evaluation

We conducted an empirical evaluation to compare the proposed method to other existing baselines. The detailed description of the used datasets and the hyper-parameter tuning procedure is given in the supplementary material.

4.1 Experimental Setup

4.1.1 Datasets

We used the following real-world datasets.

- MovieLens (ML) 1M[1]: The dataset contains five-star movie ratings collected from a movie recommendation service, and the ratings are MNAR. The dataset consists of approximately 1 million ratings from 6,040 users and 3,706 movies. In the experiments, we retained movies that are rated by at least 20 users.

[1] http://grouplens.org/datasets/movielens/

- Yahoo! R3 dataset[2]: It contains five-star user-song ratings. The training data contain approximately 300,000 MNAR ratings from 15,400 users for 1,000 songs, and the test data are collected by asking a subset of 5,400 users to rate 10 randomly selected songs.
- Coat dataset[3]: It contains five-star user-coat ratings from 290 Amazon Mechanical Turk workers on an inventory of 300 coats. The training data contains 6,500 MNAR ratings collected via self-selections by Turk workers. Conversely, the test data are collected by asking Turk workers to rate 16 randomly selected coats.

4.1.2 Train/Validation/Test Splits

For the ML 1M dataset, we created a test set with different **rating distribution** from the original one. We created it by resampling data from the test set based on the **inverse** of the rating density ratio in Eq. (12). This creates a test set with a completely different rating distribution from the training set.

$$P_r = \frac{\mathbb{P}(R = r \mid O = 1)\mathbb{P}(O = 1)}{\mathbb{P}(R = r)} \tag{12}$$

We tested the three prior rating distributions (type1, type2, and type3) for the ML 1M dataset. Type1 has a small, Type2 has a medium, and Type3 has a large difference between the training and test rating distributions[4].

For the Yahoo! R3 and Coat datasets, the original datasets were divided into training and test sets. We randomly selected 10% of the original training set for the validation set.

	ML 1M (type 1)		ML 1M (type 2)		ML 1M (type 3)		Yahoo! R3		Coat	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MF (uniform)	0.898	1.263	1.029	1.601	1.228	2.095	1.150	2.044	0.850	1.211
MF-IPS (user)	0.935	1.343	1.106	1.788	1.340	2.367	1.128	1.973	0.906	1.390
MF-IPS (item)	0.935	1.360	1.080	1.750	1.288	2.263	1.101	1.973	**0.796**	**1.065**
MF-IPS (user-item)	0.952	1.400	1.081	1.741	1.300	2.300	1.110	2.009	0.847	1.203
MF-IPS (logistic)	0.896	1.258	1.042	1.641	1.204	2.203	1.157	2.075	0.800	1.086
DAMF (ours)	**0.891**	**1.218**	**0.967**	**1.414**	**1.113**	**1.753**	**0.991**	**1.566**	0.841	1.164
MF-IPS (NB with true)	0.843	1.113	0.741	0.912	0.562	0.593	0.796	1.095	0.830	1.216

Table 1: Performance of the different approaches on all datasets over 15 different initializations. DAMF significantly outperformed the other methods on both metrics. The bold fonts represent the best performance among the methods except for the MF-IPS (NB with true). We report the model performance on test sets with the lowest validation loss out of all iterations. The results with standard deviations are given in the supplementary material.

4.1.3 Baselines & Propensity estimators

We compared the MF-IPS in [29] to our proposed DAMF. It predicts each rating by $\widehat{R}_{u,i} = \theta_u^\top \beta_i$, where $\{\theta_u\}$ and $\{\beta_i\}$ are user and item latent factors. We did not contain the use–item bias terms to yield the same

[2] http://webscope.sandbox.yahoo.com/
[3] https://www.cs.cornell.edu/~schnabts/mnar/
[4] Please see Table 3 and Figure 2 in the Supplementary Materials for detail

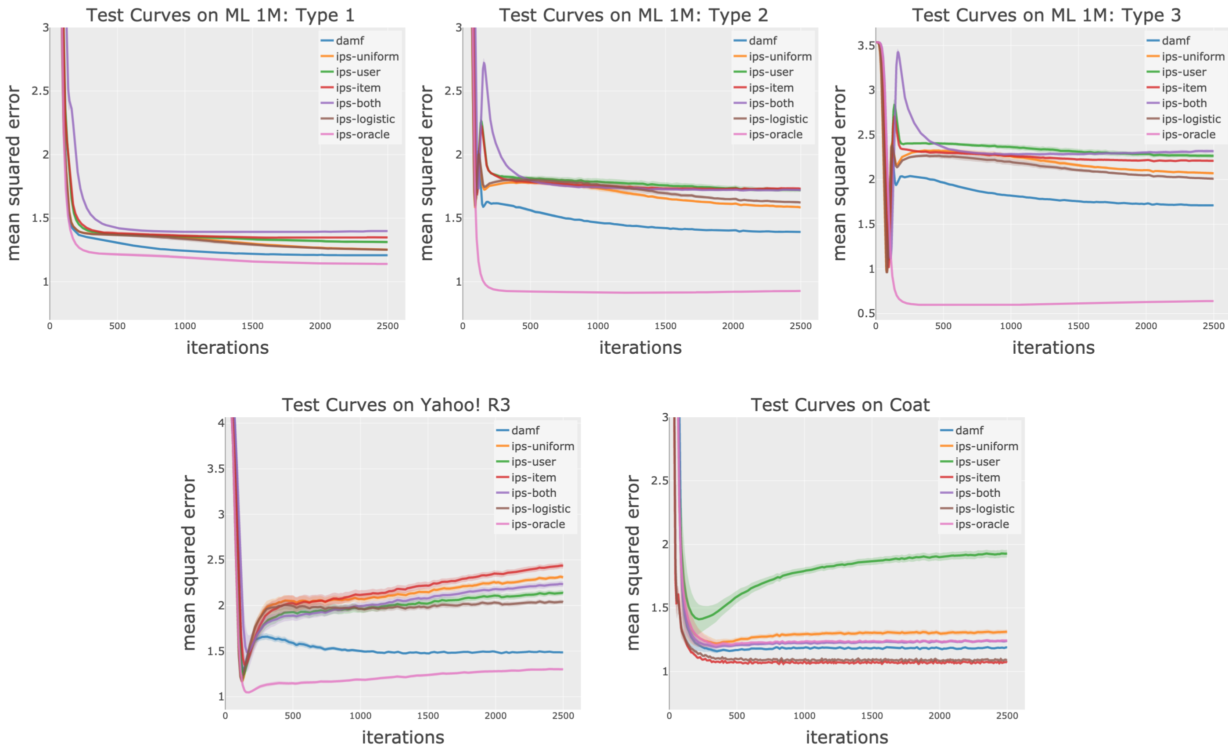

Figure 1: Test MSEs vs. number of iterations on all datasets. Note that, in Table 1, we report the model with the lowest validation loss among all iterations.

model complexity with our proposed method. The MF-IPS optimizes its parameters via minimizing the IPS loss in Eq. (3) with regularization terms.

For the MF-IPS, we tested the following propensity estimators[5].

$$\text{uniform propensity}: \widehat{P}_{*,*} = \frac{\sum_{u,i\in\mathcal{D}} O_{u,i}}{|\mathcal{D}|}$$

$$\text{user propensity}: \widehat{P}_{u,*} = \frac{\sum_{i\in\mathcal{I}} O_{u,i}}{\max_{u\in U}\sum_{i\in\mathcal{I}} O_{u,i}}$$

$$\text{item propensity}: \widehat{P}_{*,i} = \frac{\sum_{u\in\mathcal{U}} O_{u,i}}{\max_{i\in I}\sum_{u\in\mathcal{U}} O_{u,i}}$$

$$\text{user-item propensity}: \widehat{P}_{u,i} = \widehat{P}_{u,*}\cdot\widehat{P}_{*,i}$$

$$\text{logistic regression}: \boldsymbol{O}_{\text{MNAR}} \approx \hat{\boldsymbol{P}} \text{ by logistic regression}$$

It is noted that when the uniform propensity is used, the MF-IPS is identical to the MF with the naive loss function [17].

In contrast to previous studies [29, 34]; we did not use any data in the test set for the propensity estimation to imitate the real-world situation. However, in Section 4.2, we report the results with the following propensity

[5]NB represents Naive Bayes

estimator simply as reference.

$$\text{NB with true prior : } \widehat{P}_r = \frac{\mathbb{P}(R = r \mid O = 1)\mathbb{P}(O = 1)}{\mathbb{P}(R = r)}$$

where $r \in \{1, 2, 3, 4, 5\}$ is a realized rating. In reality, NB with true prior is infeasible in most real-world situations because it requires MCAR explicit feedback to estimate the prior rating distribution.

4.1.4 Hyperparameter Tuning

We tuned the dimensions of the latent factors within the range of $\{5, 10, \dots, 40\}$, and the L2-regularization parameter within the range of $[10^{-6}, 10^{-2}]$ for all methods. The trade-off hyperparameter β was tuned within the range of $[0.1, 1]$ for the proposed method. The combinations of the hyperparameters were selected using the *Optuna* software [1]. Additionally, for the proposed method, we set the top 20% frequent items in the training set as popular items and the remainder as rare.

4.2 Results & Discussions

Table 1 provides the averaged MAE and MSE on the ML 1M, Yahoo! R3, and Coat datasets.

First, consistent with the previous work [29], MF-IPS with true prior information exhibits the best performance in terms of both MAE and MSE[6]. However, MF-IPS with the other propensity estimators did not always outperform the MF-IPS with uniform propensity. The results suggest that MF-IPS is potentially an effective de-biasing method; however it is highly sensitive to the method of propensity estimation. Specifically, the use of only MNAR training data for the propensity estimation does not lead to a well-performing recommender.

Additionally, DAMF achieved significant performance gains on the three types of ML 1M and Yahoo! R3 datasets when compared to the propensity-based MF models. Furthermore, as shown in Table 1, the performance gain on the ML 1M type 3 is much larger than those on type 1 and 2. It outperformed 16.3 % in Type 3, 11.6 % in Type 2, and 3.1 % in Type 1 when compared to the best baselines. The results suggest that the benefit of the proposed method is strengthened when a large divergence exists between the training and test distributions.

Conversely, the proposed method exhibited a poor performance only on the coat dataset. This is because DAMF includes a larger number of parameters to be optimized than the baselines, and the size of this dataset is relatively small (only 6,264 ratings are in the training set). The hypothesis is also aligned with the significant gains when compared to MF-IPS in both ML 1M and Yahoo! R3 where the number of observed ratings significantly exceeds those on the coat dataset.

Figure 1 shows the Test MSEs vs. number of iterations. For the three types of ML 1M dataset, DAMF generally outperforms the MF-IPS after the 300 iterations. For the Yahoo! R3 dataset, the performance of MF-IPS first reaches a very high level and then gradually worsens with iterations. The phenomenon is similar to the memorizing effects in noisy label literature [11, 14, 21]. Conversely, DAMF alleviates the decreasing processing and almost monotonically improves its performance after 300 iterations.

In summary, the propensity-based MF models are sverely affected by the choice of propensity estimators and reveal poor performance when the true rating prior information is unavailable. Moreover, the proposed DAMF algorithm significantly outperforms the other baseline methods and especially for the moderate size

[6]The performance of the MF-IPS (NB with true) on the Yahoo! R3 data is slightly worse than that in the previous experiments [29] because we used a simple version of MF-IPS without user-item bias terms.

and severely biased datasets. The results validate the effectiveness of the proposed approach under situations where the true propensities are unknown or when costly MCAR data is unavailable.

5 Related Work

5.1 Propensity-based Recommendation

To address the bias of the MNAR explicit feedback, several related works assume the missing data model and rating model and estimate parameters using the iterative procedure [12, 22]. However, the methods are highly complex and do not perform well on real-world rating datasets [29, 35].

Propensity-based methods were proposed to solve the limitations of these converntional methods and to theoretically address the bias of MNAR feedback [20, 29, 34, 35]. Among them, the most basic means is the Inverse Propensity Score (IPS) estimation and is established in the context of causal inference [13, 24, 25]. The estimation method provides an unbiased estimator of the true metric of interest by weighting each data using the inverse of its propensity. The rating predictor based on the IPS estimator empirically outperformed the naive matrix factorization [17] and probabilistic generative model [12]. The propensity-based methods remove the bias of the naive methods although the performance of the methods mainly depend on the propensity score estimation model. Specifically, it is challenging to ensure the performance of the propensity estimator in real-world recommendations because users are free to select which items to rate, and one cannot control the missing mechanism [22]. In addition to the simple IPS estimator, [34] proposed the doubly robust (DR) variant to decrease the effect of the variance of the propensity weighting approach. The DR estimator utilizes the error imputation model and the propensity score and theoretically improves the bias and estimation error bound when compared to the IPS counterpart. However, the proposed joint learning algorithm still requires pre-estimated propensity scores [34]. Furthermore, the estimation performance of the DR estimator is significantly degraded when both error imputation models and propensity models are misspecified [15]. In the empirical evaluations of the propensity-based methods, MCAR test data is used to estimate the propensity score [29, 34]. However, in reality, the use of MCAR data is infeasible in most situations because gathering a sufficient amount of MCAR data necessitates time and cost for the annotation process.

5.2 Unsupervised Domain Adaptation

The aim of unsupervised domain adaptation (UDA) is to train a predictor that works well on a target domain by using only labeled source data and unlabeled target data during training [18, 26]. The major challenge of this field is that the feature distributions and labeling functions can differ between the source and target domains. Thus, a predictor trained using only the labeled source data does not generalize well on the target domain. Therefore, it is essential to measure the discrepancy between the two domains to achieve the desired performance on the target domain [18, 19]. Several discrepancy measures to measure the difference in the feature distributions between the source and target domains were proposed [4, 18, 19, 37]. For example, $\mathcal{H}$-divergence and $\mathcal{H}\Delta\mathcal{H}$-divergence [4, 5] were used to construct many prediction methods in UDA such as DANN, ADDA, and MCD [7, 8, 27, 33]. The methods are constructed on the adversarial learning framework and can be theoretically explained as minimizing empirical errors and discrepancy measures between the source and target domains. The optimization of these methods does not depend on the propensity score. Thus, methods of UDA are considered as beneficial in constructing an effective recommender with biased rating feedback given the absence of access to the true propensities.

The work that is most related to ours is by [2]. In this study, the propensity-independent lower bound of the performance of treatment policies are derived. The bound is based on the well-established $\mathcal{H}$-divergence and can be optimized via domain adversarial learning. The proposed policy optimization procedure empirically outperforms the propensity-based treatment policy optimization algorithm called POEM [31, 32] under the situation where the past treatment policies (propensities) are unknown. Note that the proposed method and theory in [2] are specialized to the treatment policy optimization problem and cannot be directly applied to our rating prediction problem. Our proposed method shares a structure similar to the method proposed in [2]; however ours is the first extension of the domain adversarial learning to develop a method to alleviate the bias of the MNAR recommendation without true propensity information.

6 Conclusion

In this study, we explored the problem of learning rating predictors from MNAR explicit feedback. First, we derived the generalization error bound of the loss function of interest inspired by the theoretical framework of unsupervised domain adaptation. The bound is propensity-independent; thus, problems related to the propensity estimation are eliminated in the bound. Subsequently, we proposed *Domain Adversarial Matrix Factorization* that simultaneously minimizes the naive loss of the MNAR feedback and the discrepancy between two missing mechanisms. Finally, we conducted extensive experiments on the standard real-world datasets and showed that the proposed method significantly outperformed the baseline methods under a realistic situation where the true propensities are inaccessible.

Important future research directions include the extension of the proposed method to the recommendation using implicit feedback. Furthermore, several disconnections between theory and algorithm still exist although the benefit of the proposed algorithm was empirically demonstrated. Thus, bridging the gap between the theory and algorithm is another important theme.

Acknowledgement. The author would like to thank Suguru Yaginuma and Kazuki Taniguchi for their helpful comments and discussions.

Bibliography

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA, 2019. ACM.

[2] Onur Atan, William R Zame, and Mihaela van der Schaar. Learning optimal policies from observational data. *arXiv preprint arXiv:1802.08679*, 2018.

[3] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[5] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[6] Stephen Bonner and Flavian Vasile. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 104–112, New York, NY, USA, 2018. ACM.

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[9] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. Offline a/b testing for recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 198–206. ACM, 2018.

[10] Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Frage: Frequency-agnostic word representation. In *Advances in neural information processing systems*, pages 1334–1345, 2018.

[11] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018.

[12] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. Probabilistic matrix factorization with non-random missing data. In *International Conference on Machine Learning*, pages 1512–1520, 2014.

[13] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[14] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 2017.

[15] Joseph DY Kang, Joseph L Schafer, et al. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical science*, 22(4):523–539, 2007.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[18] Seiichi Kuroki, Nontawat Charonenphakdee, Han Bao, Junya Honda, Issei Sato, and Masashi Sugiyama. Unsupervised domain adaptation based on source-guided discrepancy. *arXiv preprint arXiv:1809.03839*, 2018.

[19] Jongyeong Lee, Nontawat Charoenphakdee, Seiichi Kuroki, and Masashi Sugiyama. Domain discrepancy measure using complex models in unsupervised domain adaptation. *arXiv preprint arXiv:1901.10654*, 2019.

[20] Dawen Liang, Laurent Charlin, and David M Blei. Causal inference for recommendation. In *Causation: Foundation to Application, Workshop at UAI*, 2016.

[21] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". In *Advances in Neural Information Processing Systems*, pages 960–970, 2017.

[22] Benjamin M Marlin and Richard S Zemel. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*, pages 5–12. ACM, 2009.

[23] Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

[24] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[25] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

[26] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2988–2997, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[27] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[28] Yuta Saito, Hayato Sakata, and Kazuhide Nakata. Doubly robust prediction and evaluation methods improve uplift modeling for observational data. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 468–476. SIAM, 2019.

[29] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1670–1679, New York, New York, USA, 20–22 Jun 2016. PMLR.

[30] Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 713–722. ACM, 2010.

[31] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pages 814–823, 2015.

[32] Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. In *advances in neural information processing systems*, pages 3231–3239, 2015.

[33] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[34] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Doubly robust joint learning for recommendation on data missing not at random. In *International Conference on Machine Learning*, pages 6638–6647, 2019.

[35] Yixin Wang, Dawen Liang, Laurent Charlin, and David M. Blei. The deconfounded recommender: A causal inference approach to recommendation. *CoRR*, abs/1808.06581, 2018.

[36] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 279–287, New York, NY, USA, 2018. ACM.

[37] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413, 2019.

Supplementary Materials

A Omitted Proofs

A.1 Proof of Lemma 1

Proof. First,

$$
\begin{aligned}
&\left|d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{P},\boldsymbol{P}'\right)-d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{O},\boldsymbol{O}'\right)\right| \\
&=\left|\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}'\right)\right|-\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}'\right)\right|\right| \\
&\leq\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}\right)\right|-\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}'\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}'\right)\right|\right| \\
&\leq\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\left(\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}\right)\right)-\left(\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}'\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}'\right)\right)\right| \\
&\leq\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}\right)\right|+\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}'\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}'\right)\right|
\end{aligned}
$$

The deviations in the last line can be bounded as follows following the same logic flow in the proof of Theorem 5.2 in [29].

$$
\begin{aligned}
&\mathbb{P}\left(\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}\right)\right|\geq\epsilon\right)\leq\delta \\
&\Leftrightarrow\mathbb{P}\left(\bigcup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}\right)\right|\geq\epsilon\right)\leq\delta \\
&\Leftarrow\sum_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\mathbb{P}\left(\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}\right)\right|\geq\epsilon\right)\leq\delta \\
&\Leftarrow 2|\mathcal{H}|^{2}\exp\left(\frac{-2M^{2}\epsilon^{2}}{|\mathcal{D}|\Delta^{2}}\right)
\end{aligned}
\tag{13}
$$

Therefore, the following inequalities hold with a probability of at least $1-\delta/2$, respectively.

$$
\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}\right)\right|\leq\frac{\Delta}{M}\sqrt{\frac{|\mathcal{D}|}{2}\log\frac{4|\mathcal{H}|^{2}}{\delta}} \tag{14}
$$

$$
\sup_{\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\in\mathcal{H}}\left|\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{P}'\right)-\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{R}}'\,|\,\boldsymbol{O}'\right)\right|\leq\frac{\Delta}{M}\sqrt{\frac{|\mathcal{D}|}{2}\log\frac{4|\mathcal{H}|^{2}}{\delta}} \tag{15}
$$

Combining Eq. (13), Eq. (14), and Eq. (15) with the union bound completes the proof. □

A.2 Proof of Lemma 2

Proof. Replacing $P_{u,i}$ in Eq. (16), Theorem 5.2 in [29] for $M/|\mathcal{D}|$ completes the proof. □

A.3 Proof of Lemma 3

Proof.

$$\begin{aligned}
\widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}\right) &\leq \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{R}}^{*} \,|\, \boldsymbol{P}\right) + \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}^{*} \,|\, \boldsymbol{P}\right) \\
&\leq \left|\widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{R}}^{*} \,|\, \boldsymbol{P}\right) - \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{R}}^{*} \,|\, \boldsymbol{P}'\right)\right| + \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{R}}^{*} \,|\, \boldsymbol{P}'\right) + \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}^{*} \,|\, \boldsymbol{P}\right) \\
&\leq \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}'\right) + \sup_{\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{R}}' \in \mathcal{H}} \left|\widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{R}}' \,|\, \boldsymbol{P}\right) - \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{R}}' \,|\, \boldsymbol{P}'\right)\right| + \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}^{*} \,|\, \boldsymbol{P}'\right) + \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}}^{*} \,|\, \boldsymbol{P}\right) \\
&= \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}'\right) + d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{P}, \boldsymbol{P}'\right) + \lambda
\end{aligned}$$

□

A.4 Proof of Theorem 1

Proof. First, we obtain the following inequality by replacing $\boldsymbol{P}$ and $\boldsymbol{P}'$ for $\boldsymbol{P}_{\text{MCAR}}$ and $\boldsymbol{P}_{\text{MNAR}}$, respectively in Eq. (8).

$$\widehat{\mathcal{L}}^{\ell}_{\text{ideal}}\left(\widehat{\boldsymbol{R}}\right) \leq \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}_{\text{MNAR}}\right) + d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{P}_{\text{MCAR}}, \boldsymbol{P}_{\text{MNAR}}\right) + \lambda \tag{16}$$

where

$$\widehat{\mathcal{L}}^{\ell}_{\text{ideal}}\left(\widehat{\boldsymbol{R}}\right) = \widehat{\mathcal{L}}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}_{\text{MCAR}}\right)$$

by definition. Then, from Lemma 2 and Lemma 3, the following inequalities hold with a probability of at least $1 - \delta/2$.

$$\mathcal{L}^{\ell}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{P}_{\text{MNAR}}\right) \leq \widehat{\mathcal{L}}^{\ell}_{\text{naive}}\left(\widehat{\boldsymbol{R}} \,|\, \boldsymbol{O}_{\text{MNAR}}\right) + \frac{\Delta}{M}\sqrt{\frac{|\mathcal{D}|}{2}\log\frac{4|\mathcal{H}|}{\delta}} \tag{17}$$

$$\left|d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{P}_{\text{MCAR}}, \boldsymbol{P}_{\text{MNAR}}\right) - d_{\mathcal{H}\Delta\mathcal{H}}\left(\boldsymbol{O}_{\text{MCAR}}, \boldsymbol{O}_{\text{MNAR}}\right)\right| \leq \frac{\Delta}{M}\sqrt{2|\mathcal{D}|\log\frac{8|\mathcal{H}|^2}{\delta}} \tag{18}$$

Combining Eq. (16), Eq. (17), and Eq. (18) with the union bound completes the proof. □

B Detailed Experimental Setup

Here we describe the detailed experimental setups.

B.1 The statistics of the used datasets

The statistics of the datasets used in the experiments after preprocessing are given in Table 2. In addition, the prior rating distributions for type 1, 2, and 3 for ML 1M dataset are given in Table 3.

B.2 Hyper-parameter tuning procedure

Table 4 lists the used hyper-parameter searching spaces. Furthermore, the selected sets of hyper-parameters for all methods are listed in Table 5.

	#User	#Item	#Train data	Sparsity	Avg rating of training	Avg rating of test	KL-divergence
ML 1M (type 1)	6,040	2,836	445,705	2.26%	3.58	3.00	0.155
ML 1M (type 2)	6,040	2,836	445,705	2.26%	3.58	2.20	0.641
ML 1M (type 3)	6,040	2,836	445,705	2.26%	3.58	1.67	1.571
Yahoo! R3	15,400	1,000	280,533	1.82%	2.89	1.82	0.470
Coat	290	300	6,264	7.20%	2.61	2.23	0.049

Table 2: Statistics of datasets used in the experiments after pre-processing. KL-divergence is the divergence of rating distributions between training and test sets.

	$R = 1$	$R = 2$	$R = 3$	$R = 4$	$R = 5$
ML 1M (type 1)	0.2	0.2	0.2	0.2	0.2
ML 1M (type 2)	0.35	0.3	0.2	0.1	0.05
ML 1M (type 3)	0.5	0.4	0.05	0.03	0.02

Table 3: The test probability masses of each rating value for the three types of the ML 1M dataset.

Methods	d	λ	β	optimizer	init. learning_rate	batch_size	max iterations
MF-IPS	$\{5, 10, \ldots, 40\}$	$[10^{-6}, 10^{-2}]$	-	Adam	0.01	2^{12}	2,500
DAMF	$\{5, 10, \ldots, 40\}$	$[10^{-6}, 10^{-2}]$	$[0, 1]$	Adam	0.01	2^{12}	2,500

Table 4: Hyperparameter searching spaces. The same searching spaces were used in all datasets. Specifically, d denotes the dimension of the latent factors, and λ denotes the hyperparameter for the L2-regularization.

	ML 1M (type1)			ML 1M (type2)			ML 1M (type3)			Yahoo! R3			Coat		
Models	d	λ	β	d	λ	β	d	λ	β	d	λ	β	d	λ	β
MF-IPS (uniform)	5	9.33×10^{-6}	–	5	1.56×10^{-5}	–	5	8.24×10^{-6}	–	5	2.68×10^{-6}	–	5	9.20×10^{-4}	–
MF-IPS (user)	5	5.27×10^{-6}	–	5	1.04×10^{-5}	–	5	1.21×10^{-5}	–	5	1.38×10^{-5}	–	5	2.60×10^{-4}	–
MF-IPS (item)	5	1.85×10^{-6}	–	5	3.08×10^{-6}	–	5	2.56×10^{-6}	–	5	6.32×10^{-6}	–	40	1.38×10^{-3}	–
MF-IPS (both)	5	4.68×10^{-6}	–	5	2.49×10^{-5}	–	5	1.19×10^{-6}	–	5	1.83×10^{-5}	–	5	1.23×10^{-3}	–
MF-IPS (logistic)	5	1.07×10^{-5}	–	5	7.40×10^{-6}	–	5	1.85×10^{-5}	–	5	1.51×10^{-5}	–	30	1.30×10^{-3}	–
MF-IPS (oracle)	5	3.32×10^{-5}	–	5	1.86×10^{-5}	–	5	1.79×10^{-5}	–	5	8.99×10^{-6}	–	5	9.61×10^{-4}	–
DAMF	15	4.56×10^{-5}	0.735	15	4.56×10^{-5}	0.735	15	4.56×10^{-5}	0.735	15	4.56×10^{-5}	0.735	5	1.48×10^{-3}	0.941

Table 5: Selected sets of hyper-parameters for all methods and for all datasets.

	ML 1M (type 1)		ML 1M (type 2)		ML 1M (type 3)	
	MAE ($\pm$ SD)	MSE ($\pm$ SD)	MAE ($\pm$ SD)	MSE ($\pm$ SD)	MAE ($\pm$ SD)	MSE ($\pm$ SD)
MF (uniform)	0.898 ($\pm$0.006)	1.263 ($\pm$0.013)	1.029 ($\pm$0.006)	1.601 ($\pm$0.028)	1.228 ($\pm$0.014)	2.095 ($\pm$0.034)
MF-IPS (user)	0.935 ($\pm$0.014)	1.343 ($\pm$0.031)	1.106 ($\pm$0.011)	1.788 ($\pm$0.046)	1.340 ($\pm$0.020)	2.367 ($\pm$0.046)
MF-IPS (item)	0.935 ($\pm$0.013)	1.360 ($\pm$0.029)	1.080 ($\pm$0.018)	1.750 ($\pm$0.035)	1.288 ($\pm$0.018)	2.263 ($\pm$0.042)
MF-IPS (user-item)	0.952 ($\pm$0.009)	1.400 ($\pm$0.021)	1.081 ($\pm$0.014)	1.741 ($\pm$0.027)	1.300 ($\pm$0.012)	2.300 ($\pm$0.026)
MF-IPS (logistic)	0.896 ($\pm$0.006)	1.258 ($\pm$0.014)	1.042 ($\pm$0.010)	1.641 ($\pm$0.021)	1.204 ($\pm$0.011)	2.203 ($\pm$0.028)
DAMF (ours)	**0.891** ($\pm$0.004)	**1.218** ($\pm$0.008)	**0.967** ($\pm$0.009)	**1.414** ($\pm$0.013)	**1.113** ($\pm$0.009)	**1.753** ($\pm$0.021)
MF-IPS (NB with true)	0.843 ($\pm$0.005)	1.113 ($\pm$0.012)	0.741 ($\pm$0.006)	0.912 ($\pm$0.013)	0.562 ($\pm$0.022)	0.593 ($\pm$0.036)

Table 6: Averaged results and their standard deviations (SD) on the three types of ML 1M datasets over 15 different initializations.

	Yahoo! R3		Coat	
	MAE (± SD)	MSE (± SD)	MAE (± SD)	MSE (± SD)
MF (uniform)	1.150 (±0.018)	2.044 (±0.035)	0.850 (±0.037)	1.211 (±0.111)
MF-IPS (user)	1.128 (±0.050)	1.973 (±0.201)	0.906 (±0.046)	1.390 (±0.138)
MF-IPS (item)	1.101 (±0.001)	1.973 (±0.002)	**0.796** (±0.053)	**1.065** (±0.167)
MF-IPS (user-item)	1.110 (±0.013)	2.009 (±0.043)	0.847 (±0.030)	1.203 (±0.094)
MF-IPS (logistic)	1.157 (±0.001)	2.075 (±0.003)	0.800 (±0.030)	1.086 (±0.089)
DAMF (ours)	**0.991** (±0.014)	**1.566** (±0.027)	0.841 (±0.018)	1.164 (±0.052)
MF-IPS (NB with true)	0.796 (±0.013)	1.095 (±0.055)	0.830 (±0.020)	1.216 (±0.044)

Table 7: Averaged results and their standard deviations (SD) on the Yahoo! R3 and Coat datasets over 15 different initializations.