AD-AUG: Adversarial Data Augmentation for Counterfactual Recommendation

Yifan Wang[1†], Yifang Qin[1†], Yu Han[2], Mingyang Yin[2], Jingren Zhou[2], Hongxia Yang[2✉], and Ming Zhang[1✉]

[1] Peking University, Beijing, China
{yifanwang,qinyifang,mzhang_cs}@pku.edu.cn
[2] DAMO Academy, Alibaba Group, Hangzhou, China
{hanyu.han,hengyang.ymy,jingren.zhou,yang.yhx}@alibaba-inc.com

Abstract. Collaborative filtering (CF) has become one of the most popular and widely used methods in recommender systems, but its performance degrades sharply in practice due to the sparsity and bias of the real-world user feedback data. In this paper, we propose a novel counterfactual data augmentation framework AD-AUG to mitigate the impact of the imperfect training data and empower CF models. The key idea of AD-AUG is to answer the counterfactual question: "what would be a user's feedback if his previous purchase history had been different?". Our framework is composed of an augmenter model and a recommender model. The augmenter model aims to generate counterfactual user feedback based on the observed ones, while the recommender leverages the original and counterfactual user feedback data to provide the final recommendation. In particular, we design two adversarial learning-based methods from both "bottom-up" data-oriented and "top-down" model-oriented perspectives for counterfactual learning. Extensive experiments on three real-world datasets show that the AD-AUG can greatly enhance a wide range of CF models, demonstrating our framework's effectiveness and generality.

Keywords: Counterfactual augmentation · Collaborative filtering · Recommending systems.

1 Introduction

With an unprecedented number of products and services available on online platforms, it becomes challenging and time-consuming for users to discover interested products from overwhelming alternatives. Recommender systems have become essential tools to solve this information overloading problem by generating a personalized recommendation list for different users. Especially CF-based methods (e.g., matrix factorization[15]), which have been extensively used for recommendation, assuming users who have made similar choices tend to have similar preferences in the future.

† Both authors contributed equally to this work.
✉ Corresponding authors.

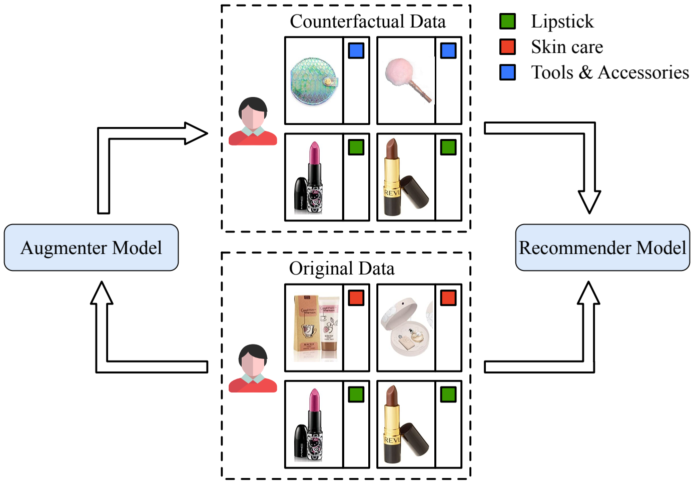


Fig. 1: An illustration of our framework for counterfactual data-augmented recommendation. The picture of each product is downloaded from https://www.amazon.com/, and the categories of the products are also presented for reference.

The core of CF methods are how to parameterize users and items with effective feature vectors based on their observed historical interactions. Notably autoencoder architecture, which has served as an effective solution for dimensionality reduction by learning underlying patterns from the interaction data between users and items, has recently applied to CF with strong performance improvements over several competitive approaches. CDAE [29] utilizes a denoising autoencoder by mapping user feedback to embeddings for reconstructing the user's preference. MultVAE [16] subsequently improves CDAE by extending model to variational autoencoders (VAE) and the likelihood to multinomial distributions. MacridVAE [18] further employs VAE to learn disentangled representations representing different user intentions.

These pure CF methods are quite efficient and effective to obtain satisfactory representations relying on the original user-item interaction data. Unfortunately, in real-world applications, most users can only access a limited number of items with large bias, as a result, pure CF can hardly capture these users' preference. Figure 1 illustrates a toy example for the product recommendation scenario. We may observe that a user has clicked Lipsticks and Skin care products influenced by the products' popularity instead of her own interest in original data. Meanwhile, although this user is in need of makeup tools and accessories, these data may not be recorded for various reasons. From the perspective of causal inference, these unrecorded data provide a key counterfactual question: “What would be a user's feedback if his previous purchase history had been different?". As a significant complementary resource of the observed user-item interactions, the counterfactual data can more comprehensively reveal the user preference.

Motivated by the above observations, we propose a novel adversarial data augmentation autoencoder model (called **AD-AUG**) for counterfactual recommendation. In general, our framework is composed of two parts (see Figure 1), an augmenter model and a recommender model. For the target user, the augmenter model generates counterfactual interaction data from the original ones, the recommender model is trained based on the original and counterfactual data to provide the final recommendation list. When building the counterfactual data from the augmenter model, we develop two types of adversarial learning-based model called data- and model-oriented methods, respectively. The data-oriented method first generates counterfactual data as different as possible from the original interaction data. Then, the model maximizes the correspondence/mutual information between the representations of the original interaction data and its augmentation in the recommendation process. While for the model-oriented method, the model adopts the principle that samples with larger loss can usually provide more knowledge to widen the experience and aims to generate counterfactual data that maximizes the information provided to the recommender model.

To summarize, in this paper we make the following contributions:

- We propose a novel adversarial training framework to empower recommendation models with counterfactual data and our framework can support a wide range of different CF models.
- We implement the above idea in two ways by developing the augmenter from both data and model perspectives to generate the counterfactual data for the recommender.
- We conduct experiments on three real-world datasets to evaluate the proposed approach. Experimental results show the effectiveness and generality of our framework.

2 Related Work

2.1 Autoencoder-based CF

Autoencoder (AE) has emerged as an important architecture to enable the CF techniques by mapping user-item interactions into latent low-dimensional representations. The goal of AEs are to minimize the reconstruction error for the user's feedback vector [23]. As the variants of AE, denoising autoencoders (DAEs) [27] and VAEs [14,22] are widely used for CF. CDAE [29] utilizes a DAE by corrupting the input feed back vector randomly. MultVAE [16] extends VAE to CF with multinomial distribution in the likelihood. MacridVAE [18] employs VAE to learn disentangled representation representing different interests of the user. RecVAE [24] proposes a new composite prior for training based on alternating updates to enhance performance. Our framework aims to design two leaning-based intervention methods to improve AE-based CF from two inseparable aspects, i.e., “data" and “model".

2.2 Counterfactual Data Augmentation

Counterfactual thinking is a concept describing the human introspection behaviors with typical question: “what would ... if ...?". It has been recently leveraged to alleviate the training data insufficiency problem in the machine learning community, e.g., computer vision [2,3,6] and natural language processing [33]. For recommendation, CASR [28] generates counterfactual user behavior sequences for sequential recommendation. To alleviate the problem of extreme sparse and imbalanced training data, CPR [32] simulates user preference and generates counterfactual samples. Instead of generating counterfactual data and make recommendation in separate two steps, our work focuses on learning counterfactual data augmentation as well as recommendation process in one union step with adversarial training.

2.3 Adversarial Training

The basic idea of adversarial training is to introduce an opponent part into the model optimization process, where two models try to detrimentally influence each other's performance and as a result, both models improve by competing against each other. Adversarial training [5] has demonstrated their abilities and potentials on a number of machine learning applications, such as image generation [12,31], language generation [17], graph representation learning [30] and robust recommender system [9]. Recently, there are some works [25] trying to combine information theory [10,26] to adversarial training. Apart from leading the training target, in this paper, we borrow the idea of adversarial training to data augmentation. The generator serves as a counterfactual data augmenter that samples challenging user interactions to optimize the recommender model.

3 Preliminaries

3.1 Problem Definition

Given a set $\mathcal{U}$ of M users and a set $\mathcal{I}$ of N items, we have a binary rating matrix $X \in \{0,1\}^{M\times N}$, where $x_{u,i} = 1$ indicates that user u explicitly adopts item i, otherwise $x_{u,i} = 0$ and it indicates a missing feedback. Given a user u, $x_u = \{x_{u,i} | i \in \mathcal{I}\}$ represents user u's history feedback vector. The goal is to learn a recommendation model $\mathcal{A}$ with x_u as input to infer user u's preference score and retrieve a ranked list of the top-N items that u prefers the most. However, accurately estimating $\mathcal{A}$ usually suffers from the data sparsity and selection bias problems. Therefore, for user u, we aim to generate sufficient “real" interaction data $\hat{x}_u$ to augment x_u.

3.2 Autoencoder CF Framework

A standard AE is trained to reproduce the input data in an output layer via a compressed latent representation. The encoder part of the framework encodes the input x_u to a d-dimensional latent representation z_u. And the decoder part of the framework takes z_u as input and outputs the reconstructed user feedback.

$$z_u = f(x_u), x'_u = g(z_u), \tag{1}$$

where $f(.)$ and $g(.)$ are encoder and decoder network respectively, which can be multiple layers neural network. And the optimization loss can be defined as,

$$\mathcal{L}_{AE}(\mathcal{A}(x_u)) = d(x'_u, x_u), \tag{2}$$

where $d(.,.)$ is the reconstruction loss. Instead of outputting the latent vectors, the encoder of VAE outputs user representation with a prior distribution. And the optimisation objective is the evidence lower bound to estimate the intractable marginal log-likelihood. For a single user u, we have:

$$\begin{aligned}&\log p(x_u) \geq \mathcal{L}_{VAE}(\mathcal{A}(x_u)) = \\ &\mathbb{E}_{z_u \sim q(z_u|x_u)} \log p(x_u|z_u) - \beta KL(q(z_u|x_u) \| p(z_u)),\end{aligned} \tag{3}$$

where KL is the Kullback-Leibler divergence distance measuring the difference between the prior distribution $p(z_u)$ and posterior distribution $q(z_u|x_u)$ parameterized by encoder function $f(.)$, $p(x_u|z_u)$ is the generated distribution conditioned on z_u parameterized by decoder $g(.)$. β is the regularization hyperparameter that balances the latent channel capacity (i.e., reconstruction accuracy) against independence constraints [10].

4 The Proposed Model

4.1 Model Overview

Inspired by the human introspection behaviors, the basic idea of our proposed model is to remove redundant information from u's positive feedback $\mathcal{I}_u^+ = \{i \in \mathcal{I} | x_{u,i} = 1\}$ and add pseudo information from missing feedback $\mathcal{I}_u^- = \{i \in \mathcal{I} | x_{u,i} = 0\}$ for the recommender learning. As shown in Figure 2 and Figure 3, beyond training a recommender model $\mathcal{A}$, our AD-AUG framework introduces an augmenter model $\mathcal{S}$ with the same model structure as $\mathcal{A}$, to answer the counterfactual question by generating counterfactual user feedback data. For the target user u, we use feed back vector x_u as input of $\mathcal{S}$, and the produced counterfactual interaction vector $\hat{x}_u$ are leveraged to optimize $\mathcal{A}$. We implement the model from both model and data perspectives, which will be introduced in the following contents.

4.2 Data-oriented Counterfactual Learning

As redundant especially incorrect user's feedback in data causes troubles for models to identify user's true preference. We introduce information bottleneck (IB) [10], which is a common practice in contrastive learning [4,30], that requests the model to capture the *minimal sufficient* information for recommendation. For the target user u, IB minimizes the information from the original feedback data x_u while maximizing the information for recommendation to remove redundant information as well as noises. As the misleading information gets removed, the model learnt by IB tends to be more robust.

Formally, we learn data-oriented counterfactual model with min-max principle,

$$\min_{\mathcal{A}} \max_{\mathcal{S}} \mathcal{L}(\mathcal{A}(x_u)) - \lambda I(\mathcal{A}_f(x_u), \mathcal{A}_f(\mathcal{S}(x_u))), \tag{4}$$

where $\mathcal{L}(\mathcal{A}(x_u))$ denotes the loss of the recommender model with x_u as input and the loss can be $\mathcal{L}_{AE}(\cdot)$ or $\mathcal{L}_{VAE}(\cdot)$ depending on the recommender model. $\mathcal{A}_f$ denotes the encoder part of the recommender model, $\mathcal{S}(x_u)$ denotes the outputs of the augmenter model, $I(\mathcal{A}_f(x_u), \mathcal{A}_f(\mathcal{S}(x_u)))$ denotes the mutual information between the original

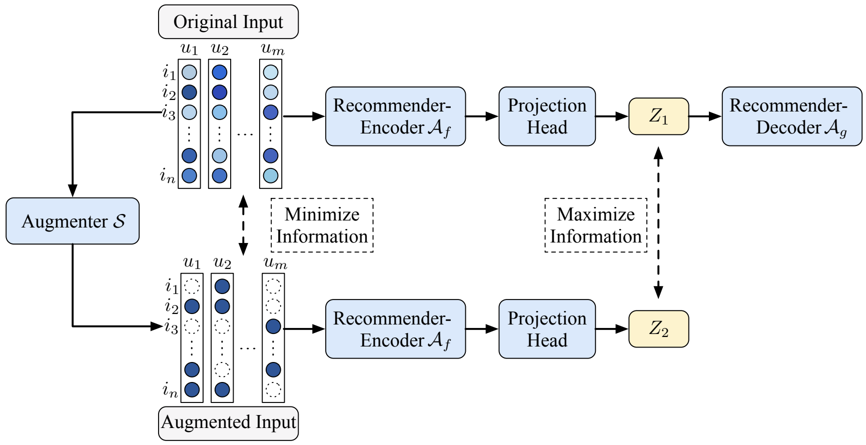


Fig. 2: The schematic view of the Data-oriented method.

user's feedback x_u and the augmented user's feedback $\hat{x}_u$, λ is the hyper-parameter to balance the mutual information and the loss.

Given the optimization target, both models are thus trained in adversarial style. During each iteration, the recommender model $\mathcal{A}$ tends to maximize the mutual information between latent representation and target labels, i.e. to minimize $\mathcal{L}(\mathcal{A}(x_u)$, while minimize the mutual information between the generated and the original data, i.e., to maximize $I(\mathcal{A}_f(x_u), \mathcal{A}_f(\mathcal{S}(x_u)))$. On the other hand, the augmenter model $\mathcal{S}$ is encouraged to generate hard negative samples, i.e., to minimize $I(\mathcal{A}_f(x_u), \mathcal{A}_f(\mathcal{S}(x_u)))$, to be distinguished from original data.

Specifically, we adopt InfoNCE as the estimator [21] for mutual information, which is known to be a lower bound of the mutual information and is frequently used for contrastive learning. Formally, during the training, given a minibatch of b users, let $z_{u,1} = h(\mathcal{A}_f(x_u))$ and $z_{u,2} = h(\mathcal{A}_f(\mathcal{S}(x_u)))$, where $h(.)$ is the projection head implemented by a 2-layer MLP as suggested in previous work [4]. We estimate the mutual information $\hat{I}$ for the minibatch,

$$\hat{I} = \frac{1}{b}\sum_{u=1}^{b} \log \frac{\exp(sim(z_{u,1}, z_{u,2}))}{\sum_{u'=1,u'\neq i}^{b} \exp(sim(z_{u,1}, z_{u',2}))}, \tag{5}$$

where $sim(.,.)$ denotes cosine similarity.

4.3 Model-oriented Counterfactual Learning

Besides augmenting user's feedback from the data perspective, we leverage the loss of the model to augment user's feedback from the model perspective. Motivated by the previous work [6,7,1], which follows the principle that samples with larger loss can usually provide more knowledge to widen the model's experience and improve the

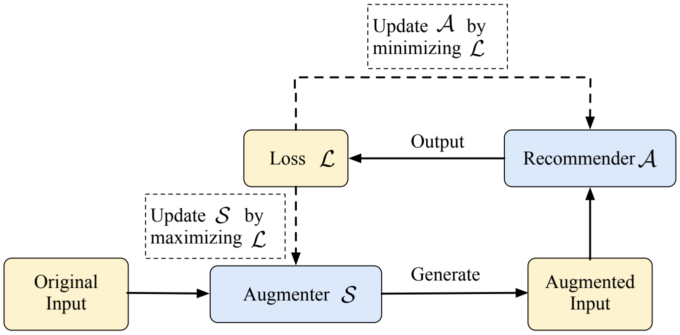

Update $\mathcal{A}$ by minimizing $\mathcal{L}$
Loss $\mathcal{L}$
Output
Recommender $\mathcal{A}$
Update $\mathcal{S}$ by maximizing $\mathcal{L}$
Original Input
Augmenter $\mathcal{S}$
Generate
Augmented Input

Fig. 3: The schematic view of the Model-oriented method.

performance. In the model-oriented method, we learn the counterfactual feedback data via maximizing the loss $\mathcal{L}$ of the recommender.

Formally, we learn the model with min-max principle defined as follows,

$$\min_{\mathcal{A}} \max_{\mathcal{S}} \mathcal{L}(\mathcal{A}(\mathcal{S}(x_u))), \tag{6}$$

where $\mathcal{L}(\mathcal{A}(\mathcal{S}(x_u)))$ denotes the loss of the recommender with augmented user's feedback $\mathcal{S}(x_u)$, i.e., $\hat{x}_u$, as input, and the loss can be $\mathcal{L}_{AE}(\cdot)$ or $\mathcal{L}_{VAE}(\cdot)$ depending on the recommender model.

4.4 Implementation of Augmenter Model

For the user's feedback data x_u, we introduce a practical instantiation of the augmenter model $\mathcal{S}$. The goal of $\mathcal{S}$ is to remove redundant information from u's positive feedback $\mathcal{I}_u^+ = \{i \in \mathcal{I} | x_{u,i} = 1\}$ and add pseudo information from missing feedback $\mathcal{I}_u^- = \{i \in \mathcal{I} | x_{u,i} = 0\}$. Specifically, each positive/missing feedback i of user u will be associated with a random variable $p_{u,i} \sim \text{Bernoulli}(\omega_{u,i})$, where $w_{u,i}$ denotes the probability of the occurrence of the interaction between u and i. For positive feedback, (u, i) is kept if $p_{u,i} = 1$ and dropped otherwise. For missing feedback, (u, i) is added as pseudo feedback in $\hat{x}_u$ if $p_{u,i} = 1$ and kept missing otherwise.

We parameterize the Bernoulli weight $\omega_{u,i}$ by leveraging another AE-based CF framework, i.e., the augmenter $\mathcal{S}$, to take x_u as input to get ω_u. In order to train $\mathcal{S}$ in an end-to-end fashion, we relax the discrete $p_{u,i}$ to be a continuous variable in $[0, 1]$ and utilize the Gumbel-Max reparameterization trick [13]. Formally,

$$p_{u,i} = \text{Sigmoid}(\frac{\log \sigma - \log(1-\sigma) + \omega_{u,i}}{\tau}), \tag{7}$$

where $\sigma \sim \text{Uniform}(0, 1)$, τ is the temperature hyper-parameter.

Meanwhile, a reasonable augmenter model $\mathcal{S}$ should keep a certain amount of information from original user feedback. Hence, we regularize the ratio of feedback being changed per user by enforcing the constraint defined as,

$$\mathcal{L}_{\text{reg}} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{i} \frac{x_{u,i}\omega_{u,i} + (1 - x_{u,i})(1 - \omega_{u,i})}{|\mathcal{I}_u^+| + |\mathcal{I}_u^-|}. \tag{8}$$

Algorithm 1: Learning Algorithm of AD-AUG

```
Input: Training user binary rating matrix $X \in \{0,1\}^{M \times N}$
Initialize: Models $\mathcal{A}$ and $\mathcal{S}$ to different initial conditions, $i \leftarrow 0$
while $i <$ MaxIteration do
    Sample one batch user feedbacks from $X$;
    Assign $\alpha$ to control changed feedback amount ;                    // Eq.11
    Learn the counterfactual user feedback $\hat{x}_u$ by $\mathcal{S}$ ;   // Eq.7
    Data-oriented Method:
    Encode $x_u$ and $\hat{x}_u$ to learn mutual information $I$;
    if $i\%2 = 0$ then
        Update $\mathcal{S}$ by maximizing $-\lambda\hat{I}+\alpha\mathcal{L}_{\text{reg}}$;
    else
        Update $\mathcal{A}$ by minimizing $\mathcal{L}-\lambda\hat{I}$ ;   // Eq.9
    end
    Model-oriented Method:
    if $i\%2 = 0$ then
        Update $\mathcal{S}$ by maximizing $\mathcal{L}+\alpha\mathcal{L}_{\text{reg}}$;
    else
        Update $\mathcal{A}$ by minimizing $\mathcal{L}$ ;                 // Eq.10
    end
    $i \leftarrow i + 1$;
end
```

4.5 Curriculum Adversarial Learning

By adding the constraint, the final objectives for the two counterfactual learning models are as follows. For data-oriented counterfactual learning, we have:

$$\min_{\mathcal{A}} \max_{\mathcal{S}} \mathcal{L}(\mathcal{A}(x_u)) - \lambda I(\mathcal{A}_f(x_u), \mathcal{A}_f(\mathcal{S}(x_u))) + \alpha\mathcal{L}_{\text{reg}}, \tag{9}$$

where α is the hyper-parameter to control the amount of user-item interaction changed from the original feedback. And for model-oriented counterfactual learning, we have:

$$\min_{\mathcal{A}} \max_{\mathcal{S}} \mathcal{L}(\mathcal{A}(\mathcal{S}(x_u))) + \alpha\mathcal{L}_{\text{reg}}. \tag{10}$$

In order to learn $\mathcal{A}$ and $\mathcal{S}$, we propose a curriculum learning method on the designed coursed, via an easy-to-difficult process. Specially, an annealing mechanism is applied:

$$\alpha = \rho * \gamma^k, \tag{11}$$

where ρ is the initial weight, γ denotes the decay ratio, and k denotes the current curriculum number. In this way, as the learned courses becomes difficult, i.e., the amount of changed feedback increase, the learned model can be gradually improved. The complete learning algorithm of our framework is shown in Algorithm 1.

Dataset	#Users	#Items	#Interactions	Sparsity
ML-1M	6,040	3,706	1,000,209	4.47%
A-Music	5,541	3,568	64,706	0.33%
A-Beauty	22,363	121,01	198,502	0.07%

Table 1: Descriptive statistics of four datasets. *Amazon-Digital Music* and *Amazon-Beauty* are simplified as *A-Music* and *A-Beauty*.

Datasets	MovieLens-1M			A-Music			A-Beauty		
Metrics	R@20	R@50	N@100	R@20	R@50	N@100	R@20	R@50	N@100
WMF	0.1072	0.1977	0.1492	0.1722	0.2534	0.1295	0.0532	0.0867	0.0435
SLIM	0.1153	0.2037	0.1589	0.1578	0.2003	0.1106	0.0166	0.0194	0.0129
CDAE	0.0929	0.1718	0.1410	0.0750	0.1366	0.0671	0.0267	0.0491	0.0232
D-CDAE	0.1044	0.1868	0.1466	0.0885	0.1541	0.0748	0.0319	**0.0571***	**0.0271***
M-CDAE	**0.1052***	**0.1891***	**0.1477***	**0.1061***	**0.1767***	**0.0872***	**0.0323***	0.0551	0.0269
MultDAE	0.1095	0.2060	0.1616	0.2021	0.3208	0.1566	0.0747	0.1224	0.0580
D-MultDAE	**0.1142***	**0.2164***	**0.1657**	**0.2160***	**0.3350***	**0.1628***	0.0784	0.1270	0.0600
M-MultDAE	0.1128	0.2114	0.1636	0.2111	0.3326	0.1603	**0.0791***	**0.1288***	**0.0611***
MultVAE	0.1132	0.2142	0.1659	0.2062	0.3241	0.1579	0.0782	0.1245	0.0588
D-MultVAE	0.1167	0.2169	0.1681	0.2178	**0.3398***	**0.1648***	0.0786	0.1281	0.0605
M-MultVAE	**0.1180**	**0.2196***	**0.1697***	**0.2192***	0.3354	0.1643	**0.0793**	**0.1294***	**0.0609***
MacridVAE	0.1130	0.2167	0.1658	0.2413	0.3626	0.1803	0.1036	0.1559	0.0753
D-MacridVAE	0.1176	**0.2220***	0.1691	**0.2485***	**0.3716***	**0.1861***	0.1082	0.1642	0.0790
M-MacridVAE	**0.1185***	0.2215	**0.1707***	0.2478	0.3699	0.1844	**0.1087***	**0.1652***	**0.0796***

Table 2: Results of effectiveness experiments on four different datasets. We use “D-X" and “M-X" to represent the data- and model-oriented counterfactual learning when the backbone model is “X". Statistical significance of pairwise differences of AD-AUG vs. the backbone model is determined by a paired t-test (* for $p \leq 0.01$).

5 Experiment

5.1 Experimental Settings

Datasets. We validate the proposed framework on three public available datasets. In specific, *MovieLens* is a widely used benchmark dataset in movie recommendation, we conduct experiments on a widely used subset of this dataset, *MovieLens-1M*. *Amazon* is a widely used benchmark dataset for product recommendation [8]. We select *Digital Music* and *Beauty* subsets from the collection. For each dataset, we treat each review as an interaction between the user and item to transform the it into implicit data. The statistics of the datasets are summarized in Table 1.

MovieLens: https://grouplens.org/datasets/movielens/
Amazon: http://jmcauley.ucsd.edu/data/amazon/

Baselines. To demonstrate the effectiveness, we compare AD-AUG with the following representative models.

- **WMF** [11] is a weighted matrix factorization method, which decomposes the implicit user feedback similar to SVD but with confidence weights defined as number of times user interacted with item.
- **SLIM** [20] learns a sparse matrix of aggregation coefficient that corresponds to the weight of rated items aggregated to produce recommendation scores.
- **CDAE** [29] is an AE-based CF model which uses a demonising autoencoder for recommendation.
- **MultDAE** [16] extends CDAE by using a multinomial likehood for the data distribution.
- **MultVAE** [16] is a VAE-based CF model which uses a multinomial likelihood for VAE to improve the recommendation performance.
- **MacridVAE** [18] employs VAE to learn disentangled representation representing different interests of the user.

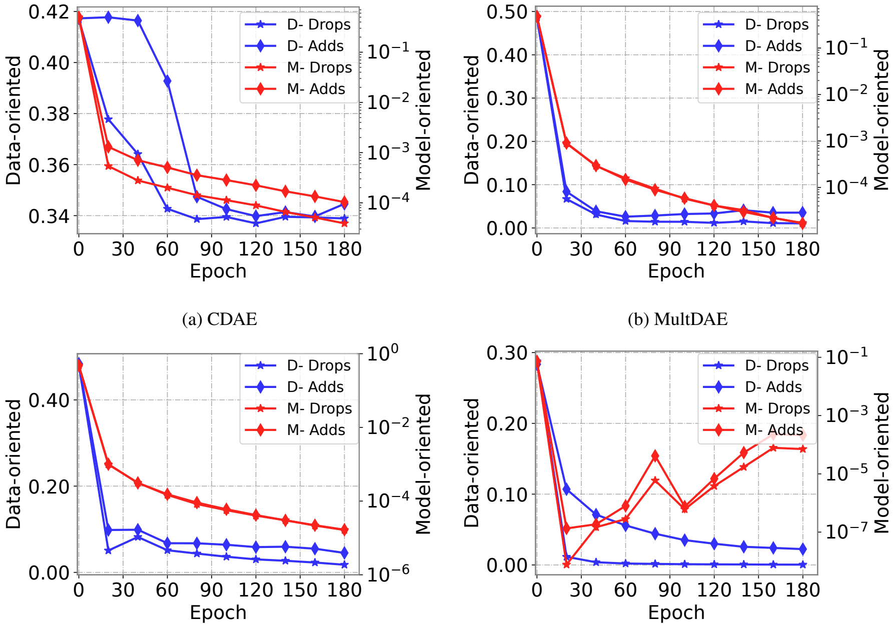
Evaluation Metrics. For each user of the dataset, we rank the interactions in chronological order and select the first 80% of historical interactions as the training set with the remaining 10%, 10% as the validation and test set respectively. For testing, we regard all unrated items as candidates and employ three metrics, *Recall (R)@K* with $K \in \{20, 50\}$ and *Normalized Discounted Cumulative Gain (NDCG or N)@K* with $K = 100$, which are computed based on rank of test interactions in top-K ranked list.

Implementation Detail. We implement our AD-AUG in Pytorch. The embedding size of user representation is fixed to 100 for all experiments. The encoder and decoder consist of two layers with $[500, 300]$ and $[300, 500]$ respectively, each with ReLU activation. For our method, the hyper-parameter $\tau = 1.0$, $\beta = 0.2$, $\gamma = 0.99$, and dropout with probability $p = 0.5$ is employed to the input. We set $\lambda = 1.0$ and the initial value $\rho = 10$ for data oriented method while $\rho = 1000$ for model oriented method in curriculum adversarial learning. We optimize AD-AUG with Adam optimizer with the learning rate as 0.001 to both augmenter and recommender and using early stopping with a patience of 50, i.e. we stop training if NDCG@100 on the validation set does not increase for 50 successive epochs. For baseline methods, we split exactly the same training, validation and test set as AD-AUG and apply a gird search for optimal hyper-parameters.

5.2 Experimental Result

Overall Comparison. We summarize the results by comparing the performance of all the methods. As shown in table 2, MacridVAE performs best among all the AE-based CF methods, which demonstrates the effectiveness of disentangled representation modeling different user interests for recommendation. Meanwhile, compared with these baselines, the data- and model-oriented models have achieved a significant improvement over all datasets, especially model-oriented method. This is actually not surprising, since the counterfactual user feedback generated from the model-oriented method is

The implementations are available at https://github.com/Fang6ang/AD-AUG



(a) CDAE

(b) MultDAE

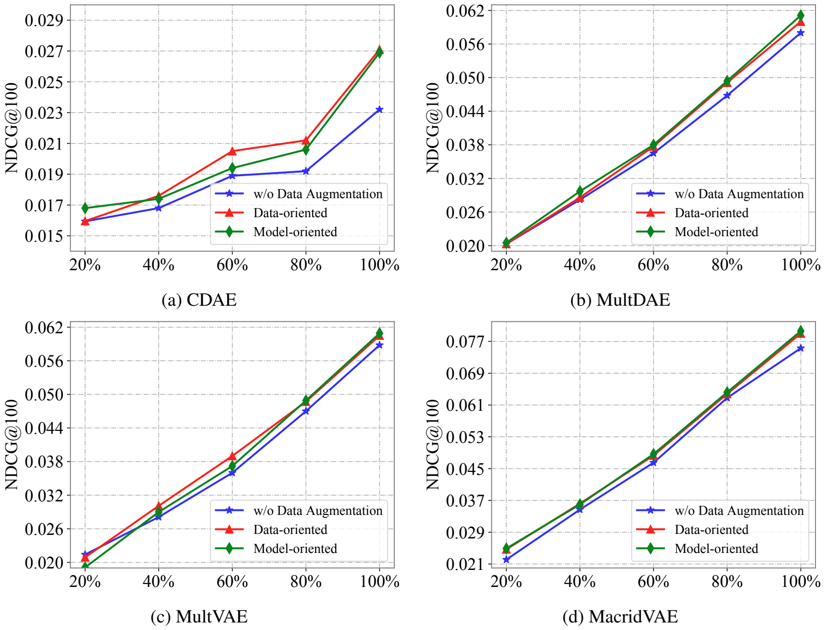
(c) MultVAE

(d) MacridVAE

Fig. 4: Training dynamics of drop/add ratio in augmented data on *Amazon-Beauty*.

more targeted, which is tailored for improving the recommender. However, the model-oriented method is achieved by using the information of the recommender, when the loss of the recommender model is unavailable (as black box for augmenter), it is better to use data-oriented method.

The Statistics of Data Augmentation. As there are two proposed methods of data augmentation, we study the augmentation results during whole training process respectively. The ratio of interactions changed by augmenter model is recorded during training process under the sets of hyper-parameters where each model achieves its best performance. As shown in Figure 4, the ratio of generated counterfactual interactions decrease until they converge during training process. Meanwhile, the ratio differs between data- and model- oriented methods during the training process. The change ratio of data-oriented method is significantly higher than model-oriented method on each epoch. On the one hand, this observation indicates that data-oriented method, which leverages mutual information to augment training data, would prefer more changes on the interaction data to assist the recommender to extract more information from training data. On the other hand, the fewer change ratio of model-oriented method means a few interactions are added or dropped by the augmenter to reach the best performance. This is because that original user feedback is sparse, and this causes the number of adding/deleting user interactions from the original feedback small for the model-oriented method.



(a) CDAE

(b) MultDAE

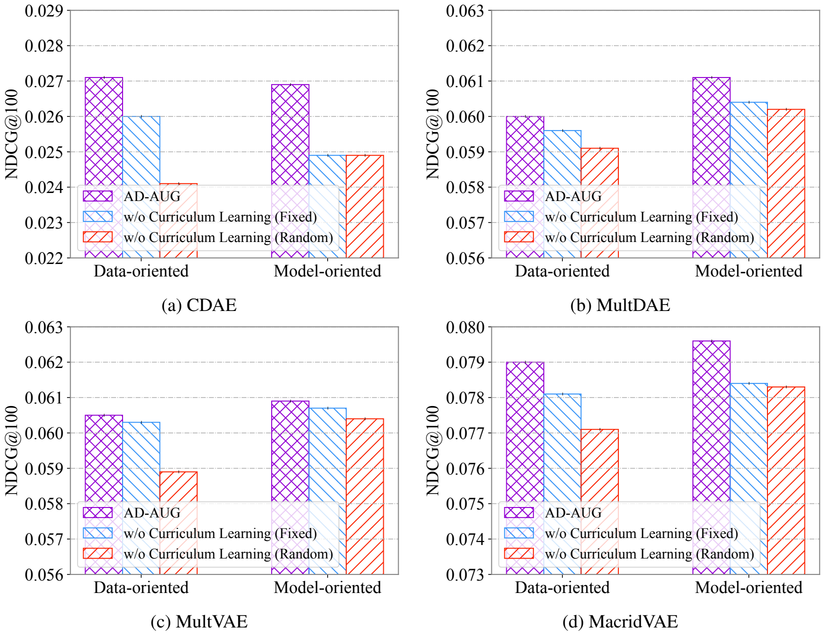
(c) MultVAE

(d) MacridVAE

Fig. 5: Performance comparison over the sparsity distribution of data on *Amazon-Beauty*.

The Effect of Data Augmentation. The number of user feedback is an important factor that affects the recommendation performance since fewer user-item interactions are insufficient to generate high-quality representations. We study whether our data augmentation methods can alleviate this sparsity issue. Towards this end, we divide the feedback data of user in the training data into five equal folds and vary the amount of training data from one fold to four folds, corresponding to 20%, 40%, 60%, 80% of entire training data as training sets. Figure 5 illustrates the performance w.r.t. different sparsity distribution of data on *Amazon-Beauty*, the performance substantially drops when less training data is used. Meanwhile, we can see our data- and model-oriented counterfactual learning can enhance the performance of each AE-based CF models, and the improvements are particularly significant when the user feedback is relatively sparse (40% to 80% of user feedback). The result indicates that AD-AUG helps improve recommendation for inactive users by generating the counterfactual user feedback.

Influence of Curriculum Learning. To investigate how the curriculum adversarial learning affects the performance, we compare the adversarial learning process under three different annealing configurations: our complete method, our model without annealing that set α with fixed initial value ρ (Fixed), our model that randomly set α between 0 to ρ under each curriculum step (Random). As shown in Figure 6, for the model-oriented counterfactual learning, the performance has slightly been affected without the curriculum learning. In contrast, curriculum learning has a more significant effect on data-oriented counterfactual learning. The best results are attained by considering the



(a) CDAE

(b) MultDAE

(c) MultVAE

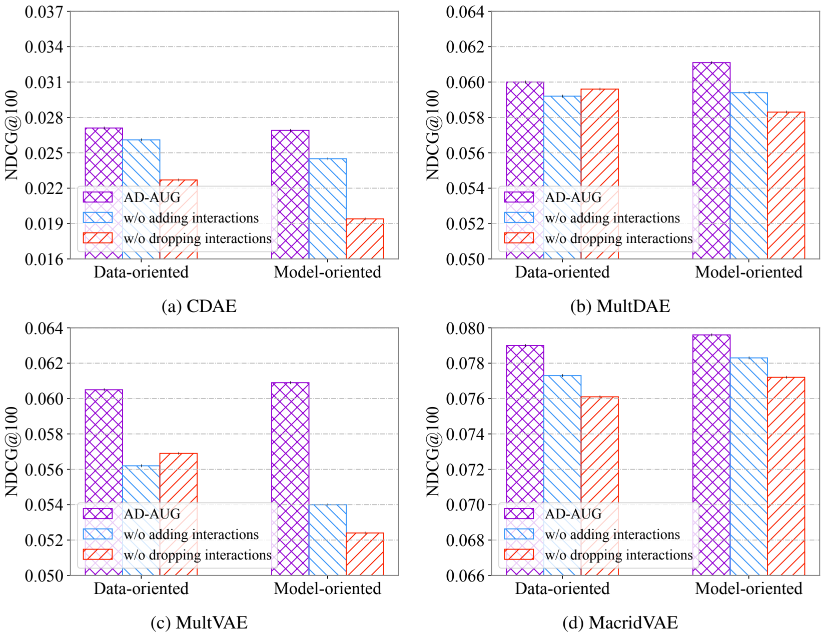
(d) MacridVAE

Fig. 6: Effect of curriculum adversarial learning on *Amazon-Beauty*.

easy-to-difficult curriculum pattern, which indicates that curriculum adversarial learning makes the objective smoother, thus more easily reaching the global optimal.

Ablation Study. We conduct an ablation study to verify the effectiveness of the two proposed data augmentation methods. Figure 7 compares the recommender model's performance on *Amazon-Beauty* when the augmenter model applies only one kind of change to the original data, i.e., adding or dropping interactions in user's behavior history. As shown in Figure 7, model's performance declines when there is only one kind of change is applied to the interaction data. The results indicate that both adding and dropping is required in data augmentation: adding interactions helps the recommender discover implicit feedback from original data, while dropping interactions removes noises from observed interaction history. The model performance suffers more under w/o drops settings. We suspect that it is because data denoising plays a more fundamental role for recommender model in terms of discovering user's true interests, while just adding interactions may lead to more noisy data.

Visualization and Case Study. To further investigate how our counterfactual data augmentation framework facilitates the user representation learning, we visualize the learnt hidden representations and conduct case study on *Amazon-Beauty*. We use MacridVAE as backbone and visualize the high-dimensional user representations learned by MacridVAE and our model. Then we randomly select a user (67) and present this user's Top-3 add/delete products for the original feedback. As shown in Figure 8, we treat each



(a) CDAE

(b) MultDAE

(c) MultVAE

(d) MacridVAE

Fig. 7: Ablation study about different types of data augmentation on *Amazon-Beauty*.

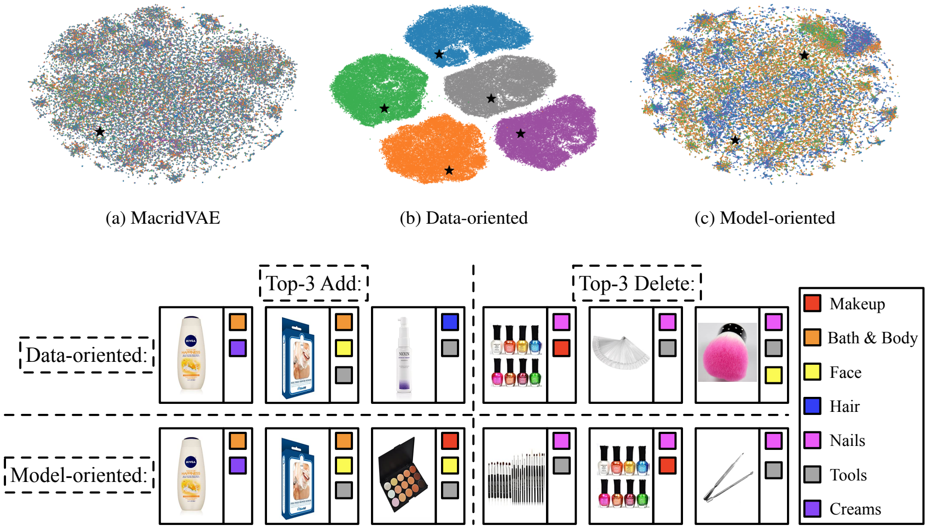
learned disentangle component of a user as an individual point and the k-th component is colored according to k. Compared with backbone model, the data- and model-oriented learning frameworks show different cluster structures, especially data-oriented method, which can form clearer clusters. Additionally, as the random selected user(67) mostly focus on nail makeups and relevant tools, the two counterfactual frameworks remove some of the nail tools and add bath equipment and other makeup products to the feedback. It indicates that the counterfactual data makes the model learned more personalized characteristics.

6 Conclusion

In this paper, we propose to improve CF performance by enriching the user feedback based on the idea of counterfactual thinking. To achieve goal, we design two adversarial learning-based data augmentation methods to generate the counterfactual user feedback data for recommendation. Experiments demonstrate that our proposed AD-AUG model achieves considerable improvement compared with state-of-the-art models.

7 Acknowledgments

This research is partially supported by National Key Research and Development Program of China with Grant No. 2018AAA0101902, the National Natural Science Foundation of China (NSFC Grant 62106008 & 62006004).



(a) MacridVAE

(b) Data-oriented

(c) Model-oriented

(d) Case study

Fig. 8: Visualization of the learned t-SNE [19] transformed user representations on *Amazon-Beauty*, where the marked stars represent a user (67). And Figure 8d represents the case study of this user.

References

1. Abbasnejad, E., Teney, D., Parvaneh, A., Shi, J., Hengel, A.v.d.: Counterfactual vision and language learning. In: CVPR. pp. 10044–10054 (2020)
2. Ashual, O., Wolf, L.: Specifying object attributes and relations in interactive scene generation. In: ICCV. pp. 4561–4569 (2019)
3. Chen, L., Zhang, H., Xiao, J., He, X., Pu, S., Chang, S.F.: Counterfactual critic multi-agent training for scene graph generation. In: ICCV. pp. 4613–4623 (2019)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607 (2020)
5. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: An overview. IEEE Signal Processing Magazine **35**(1), 53–65 (2018)
6. Fu, T.J., Wang, X.E., Peterson, M.F., Grafton, S.T., Eckstein, M.P., Wang, W.Y.: Counterfactual vision-and-language navigation via adversarial path sampler. In: ECCV. pp. 71–86. Springer (2020)
7. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual visual explanations. In: ICML. pp. 2376–2384. PMLR (2019)
8. He, R., McAuley, J.: Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In: WWW. pp. 507–517 (2016)
9. He, X., He, Z., Du, X., Chua, T.S.: Adversarial personalized ranking for recommendation. In: SIGIR. pp. 355–364 (2018)
10. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: ICLR (2017)

11. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: ICDM. pp. 263–272 (2008)
12. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. pp. 1125–1134 (2017)
13. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: ICLR (2017)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
15. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer **42**(8), 30–37 (2009)
16. Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: WWW. pp. 689–698 (2018)
17. Lin, K., Li, D., He, X., Zhang, Z., Sun, M.T.: Adversarial ranking for language generation. In: NeuIPS. pp. 3155–3165 (2017)
18. Ma, J., Zhou, C., Cui, P., Yang, H., Zhu, W.: Learning disentangled representations for recommendation. In: NeuIPS. pp. 5712–5723 (2019)
19. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
20. Ning, X., Karypis, G.: Slim: Sparse linear methods for top-n recommender systems. In: ICDM. pp. 497–506 (2011)
21. Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., Tucker, G.: On variational bounds of mutual information. In: ICML. pp. 5171–5180 (2019)
22. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: ICML. pp. 1278–1286. PMLR (2014)
23. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: Autorec: Autoencoders meet collaborative filtering. In: WWW. pp. 111–112 (2015)
24. Shenbin, I., Alekseev, A., Tutubalina, E., Malykh, V., Nikolenko, S.I.: Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In: WSDM. pp. 528–536 (2020)
25. Suresh, S., Li, P., Hao, C., Neville, J.: Adversarial graph augmentation to improve graph contrastive learning. pp. 15920–15933 (2021)
26. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? In: NeurIPS. pp. 6827–6839 (2020)
27. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: ICML. pp. 1096–1103 (2008)
28. Wang, Z., Zhang, J., Xu, H., Chen, X., Zhang, Y., Zhao, W.X., Wen, J.R.: Counterfactual data-augmented sequential recommendation. In: SIGIR. pp. 347–356 (2021)
29. Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising auto-encoders for top-n recommender systems. In: WSDM. pp. 153–162 (2016)
30. Xu, D., Cheng, W., Luo, D., Chen, H., Zhang, X.: Infogcl: Information-aware graph contrastive learning. In: NeurIPS. pp. 30414–30425 (2021)
31. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: CVPR. pp. 1316–1324 (2018)
32. Yang, M., Dai, Q., Dong, Z., Chen, X., He, X., Wang, J.: Top-n recommendation with counterfactual user preference simulation. In: CIKM. pp. 2342–2351 (2021)
33. Zmigrod, R., Mielke, S.J., Wallach, H., Cotterell, R.: Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. arXiv preprint arXiv:1906.04571 (2019)