Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01547-4

ORIGINAL ARTICLE

# A counterfactual explanation method based on modified group influence function for recommendation

Yupu Guo[1] · Fei Cai[1] · Zhiqiang Pan[1] · Taihua Shao[1] · Honghui Chen[1] · Xin Zhang[1]

Received: 24 February 2024 / Accepted: 29 May 2024


**Abstract**
In recent years, recommendation explanation methods have received widespread attention due to their potentials to enhance user experience and streamline transactions. In scenarios where auxiliary information such as text and attributes are lacking, counterfactual explanation has emerged as a crucial technique for explaining recommendations. However, existing counterfactual explanation methods encounter two primary challenges. First, a substantial bias indeed exists in the calculation of the group impact function, leading to the inaccurate predictions as the counterfactual explanation group expands. In addition, the importance of collaborative filtering as a counterfactual explanation is overlooked, which results in lengthy, narrow, and inaccurate explanations. To address such issues, we propose a counterfactual explanation method based on Modified Group Influence Function for recommendation. In particular, via a rigorous formula derivation, we demonstrate that a simple summation of individual influence functions cannot reflect the group impact in recommendations. After that, building upon the improved influence function, we construct the counterfactual groups by iteratively incorporating the individuals from the training samples, which possess the greatest influence on the recommended results, and continuously adjusting the parameters to ensure accuracy. Finally, we expand the scope of searching for counterfactual groups by incorporating the collaborative filtering information from different users. To evaluate the effectiveness of our method, we employ it to explain the recommendations generated by two common recommendation models, i.e., Matrix Factorization and Neural Collaborative Filtering, on two publicly available datasets. The evaluation of the proposed counterfactual explanation method showcases its superior performance in providing counterfactual explanations. In the most significant case, our proposed method achieves a 17% lead in terms of Counterfactual precision compared to the best baseline explanation method.

**Keywords** Recommender systems · Counterfactual explanations · Group influence functions

## Introduction

Recommender systems (RS) have become an indispensable part of e-commerce [1], social media [2–4] and online content platforms [5]. These systems provide personalized services to users, improving users' satisfaction and loyalty. Consequently, extensive researches [2, 6–8] focus on improving the performance of recommender systems for predicting user behavior, benefiting businesses and facilitating users' information access. Among them, the neural recommendation models have gained significant advantages and emerged as a major frontrunner. However, due to the complexity and black-box nature of neural recommendation models, users often have doubts and mistrust regarding the explainability of recommendations. To overcome this obstacle, interpretable recommendation [9, 10] and explanation methods [11–13] have gained constant attention from both academic and industry communities.

In the typical explanation methods, the additional auxiliary information such as attributes [14–16], reviews [13, 17], and knowledge graphs [18–20] are often used to facilitate the explanations of recommendations. However, obtaining such information in practical scenarios is not only challenging but controversial [13, 21]. In most practical situations, only the interactions are available for generating recommendations and explanations. In such cases, the influence functions [22] are deemed useful for constructing the counterfactual explanations. Specifically, the influence functions evaluate how the model parameters will change after removing a training sample, addressing the counterfactual question: "How

✉ Fei Cai
18686073813@163.com

[1] National University of Defense Technology, Changsha, China

Published online: 27 July 2024

would the model parameters change if a training sample were actually absent during training?" [23]. For instance, [24] establish the relationship between the model parameters and the prediction scores, which calculates the influence of a particular training sample on the prediction score. Building upon this work, [25] propose a rough counterfactual explanation method named ACCENT for top-N recommendation, which identifies a counterfactual group from the training set based on the sum of their influence functions. The counterfactual group can be considered as an explanation, because removing the training samples in the group changes the recommended results.

However, previous studies did not theoretically derive the group influence function, such as ACCENT using a simple summation of individual influence functions to represent the group influence function, which may inadequately capture the group influence on the model parameters, as demonstrated in "Group influence function". Our analysis reveals that a small quantity overlooked in the derivation of the individual influence function will become significant when the counterfactual group is sizable, particularly in the scenarios where the training sample set is not exceptionally large. Additionally, we observe that the collaborative filtering information is often ignored, e.g., ACCENT [25], indicating that the models solely leverage the target user's historical interactions as candidate counterfactual samples while disregarding other users' interactions [25]. For instance, in Fig. 1, the objective of the counterfactual explanation method is to identify a counterfactual group from the pertinent training samples, such that the removal of this group from the training set alters the recommended results. These pertinent training samples will encompass not only the interaction records of the target user $A$, but the interaction records of other users with the top item "Avatar 2" (e.g., $B$ and $C$), as they influence the embedding generation of "Avatar 2" during the training process.

To address the aforementioned issues, we propose a **M**odified counterfactual **G**roup selection method based on **I**nfluence **F**unction, i.e., MGIF, which effectively identifies the precise and concise counterfactual groups providing the explanations for recommendations. We initially derive the expression of the group influence function and find it impossible to directly determine a counterfactual group all at once. Therefore, MGIF innovatively iterate through the training samples to find and add samples to the counterfactual group, continuously adjusting parameters during this process to ensure accuracy. Moreover, we constantly adjust the model parameters in this process to take into account as much as possible the small quantities ignored by the individual influence function. Furthermore, we incorporate the collaborative filtering interactions in the process of identifying the counterfactual groups. As in Fig. 1, the explanation produced by MGIF can encompass not only "you liked" but "similar others liked".

In general, the major contributions in this paper are summarized as follows.

1. We prove that the group influence function should not be represented by the simple sum of individual influence functions when generating counterfactual explanations for recommendation and propose a modified group influence function that can accurately express group influence;
2. Based on the modified group influence function, We devise a counterfactual explanation algorithm for searching the counterfactual group, which considers the collaborative filtering information of other users;
3. We verify the efficacy and viability of the proposed method through extensive experiments on two public datasets.

In summary, the theoretical significance of our work lies in demonstrating that in the field of recommendation, the removal of a group cannot be simply obtained by adding up individual influence functions. The practical significance of our work is that we have greatly improved the accuracy of counterfactual explanations in recommendation, which can help recommendation systems provide explanations to users without the need for repetitive training and validation. The remaining sections of this paper are organized as follows: "Related work" provides an overview of the relevant works, encompassing the current state of research on interpretability of recommender systems, as well as the influence function. Next, "Method" presents a comprehensive exposition of the proposed MGIF method. After that, the efficacy and viability of the proposed method are meticulously evaluated in "Experiments". Finally, in "Conclusion", we summarize the main work of this paper and the perspectives of future research.

## Related work

### Explainable recommendation

In recent years, due to the rapid advancement of artificial intelligence and machine learning, significant progress has been made in various application domains such as image processing, natural language processing, and more. However, as the majority of machine learning models are black box models, achieving impressive results but lacking interpretability, the need for explainable artificial intelligence [26–28] has garnered increasing attention from both academia and industry. Among these, the Explainable recommender systems has become a crucial research topic [29, 30].

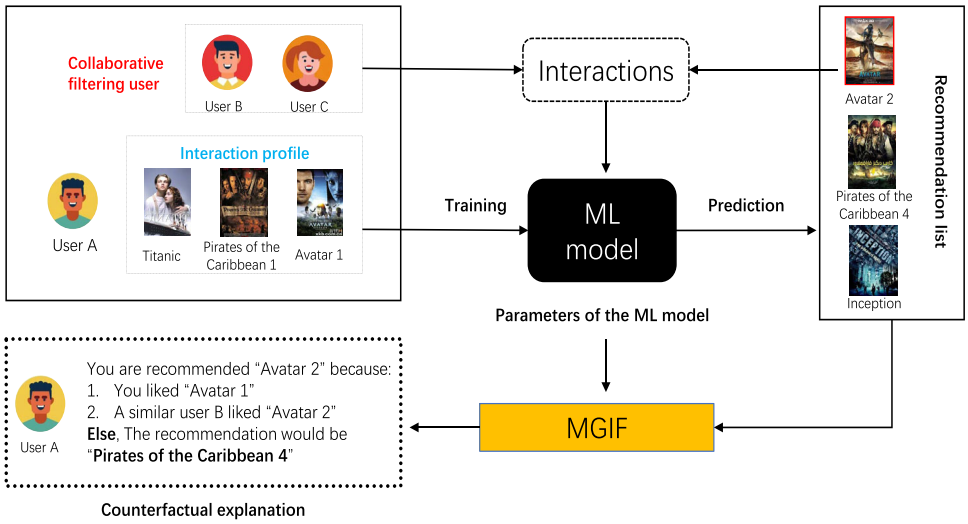


**Fig. 1** Examples of counterfactual explantion for recommendations

Explainable recommender systems provide explanations to clarify why certain items are recommended. This transparency improves the effectiveness, persuasiveness, and trustworthiness of the recommender systems[17, 31]. Explanations can be presented in various formats, such as ranked sentences[32, 33], latent factor extraction [34], knowledge graph paths [35–37], reasoning rules [38, 39] as well as natural language explanations [29, 30] and model-agnostic explanations [40]. Although the above methods have become increasingly popular due to their user-friendly interpretation, they heavily rely on auxiliary information such as textual data, knowledge graphs, and rule-based information. However, recommender systems often only have access to interaction data in practical applications, and such auxiliary information is often lacking or unavailable.

In order to provide explanations for recommendations in the absence of auxiliary information, [25] are inspired by the concept of influence functions and propose a framework for counterfactual explanation. They achieve this by continuously identifying training samples that, when removed, would favor the candidate item over the target item, thereby forming counterfactual groups to explain the recommendations. In summary, counterfactual explanations primarily address a counterfactual question: "If certain samples were missing during training, would the model recommend the candidate item instead of the current item?" However, the first-order influence function [24] measures the influence of the absence of a specific training sample on the prediction results, and its simple summation does not accurately represent the collective influence of the absence of a sample group on the results. To investigate the group influence, [23] leverage perturbation theory and introduce a second-order influence function, which makes significant progress in predicting the group counterfactual influence compared to the first-order influence function. However, their work primarily focuses on calculating the influence of a given group on the prediction results and does not provide a solution for finding the counterfactual group, failing to meet our requirement of providing counterfactual explanations for recommendations. To bridge this gap, in this paper, we present an modified method for finding counterfactual groups.

## Influence function

Influence function is a classic technique from robust statistics and used to help explain the prediction of black-box models. Essentially, the influence function predicts how the model parameters will change if a training sample is removed during training. Influence function is expressed as follows:

$$I(z) = \hat{\theta} - \hat{\theta}^{-z} = -\frac{1}{n} H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta}) \tag{1}$$

where $\hat{\theta}$ is the optimal parameters obtained after training, $\hat{\theta}^{-z}$ represents the optimal parameters of the model that we expect to get after removing the training sample $z$ and retrain the model. $n$ is the number of the total training samples. $H_{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta}^2 L(z, \hat{\theta})$ is the hessian matrix and $\nabla_{\theta} L(z, \hat{\theta})$ is the derivate of the loss at $z$ with respect to $\hat{\theta}$. It is noticeable that the conclusion of the influence function involves an important mathematical conclusion:

$$\frac{d\hat{\theta}_{\epsilon}}{d\epsilon} = -H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta}), \tag{2}$$

which we deduce again in "Group influence function" to show that the group influence function cannot be obtained simply by adding the individual influence function.

On the basis of influence function on model parameters, [24] deduce the influence of a certain training sample on the recommended predicted score $\hat{y}$ as follows:

$$\begin{aligned} INFL(z) &\stackrel{def}{=} \hat{y}(x,\hat{\theta}) - \hat{y}(x,\hat{\theta}^{-z}) \\ &= -\frac{1}{n}\nabla_\theta \hat{y}(x,\hat{\theta}) H_{\hat{\theta}}^{-1} \nabla_\theta L(z,\hat{\theta}), \end{aligned} \tag{3}$$

where $\hat{y}(x,\hat{\theta})$ is the model's prediction on the test point $(x, y)$ with parameters $\hat{\theta}$, $x$ is input and $y$ is output. Then, following Eq. (3), the influence of the training sample $z$ on the score gap between two items $i$ and $j$ can be estimated as follows:

$$\begin{aligned} INFL(z, \hat{y}(x_i) - \hat{y}(x_j)) &= (\hat{y}(x_i,\hat{\theta}) - \hat{y}(x_j,\hat{\theta})) \\ &\quad - (\hat{y}(x_i,\hat{\theta}^{-z}) - \hat{y}(x_j,\hat{\theta}^{-z})) \\ &= (\hat{y}(x_i,\hat{\theta}) - \hat{y}(x_i,\hat{\theta}^{-z})) \\ &\quad - (\hat{y}(x_j,\hat{\theta}) - \hat{y}(x_j,\hat{\theta}^{-z})) \\ &= INFL(z, \hat{y}(x_i)) \\ &\quad - INFL(z, \hat{y}(x_j)) \end{aligned} \tag{4}$$

Based on the above conclusion, the Accent method [25] approximates the influence of removing a set $Z = \{z_1, z_2, ..., z_n\}$ as the sum of individual influence function:

$$INFL(Z, \hat{y}(x_i) - \hat{y}(x_j)) = \sum_{k=1}^{n} INFL(z_k, \hat{y}(x_i) - \hat{y}(x_j)) \tag{5}$$

From the above work, the influence function emerges as a convenient tool. It utilizes the model parameters and gradients acquired during training to estimate the prediction outcome when a training sample is excluded. Consequently, it eliminates the need for retraining the model without the specific training sample, saving time and effort. Essentially, the influence function addresses the hypothetical query: "What changes would occur in the model parameters and results if a particular training sample were removed?"

## Method

In this section, we first provide a derivation of the group influence function in "Group influence function" and explain why it is not accurate to represent the group influence function by simply summing up the single influence in the field of recommender systems. Subsequently, we elaborate on the proposed method called MGIF in "Counterfactual group searching method based on modified group influence function", which aims to accurately seek counterfactual explanations for recommendations. The symbol definitions used in this paper are presented in Table 1.

### Group influence function

Based on the individual influence function introduced in "Influence function", we hope to find a group influence function to calculate the influence of removing a group of training samples $U = \{z_1, z_2, ..., z_u\}$ on the model and recommendation results.

Assume that training a model on a dataset is achieved by optimizing the overall loss function as follows:

$$L_\phi(\theta) = \frac{1}{|S|}\sum_{z\in S} L(z,\theta), \tag{6}$$

where $L(z,\theta)$ is the loss of the training sample $z$, and $S$ is the set of total related training samples. If we train the model with the set $U$ removed from the dataset $S$, the loss function becomes as follows:

$$L_U(\theta) = \frac{1}{|S| - |U|}\sum_{z\in S/U} L(z,\theta). \tag{7}$$

The essence of the influence function is to estimate the parameters after removing the set $U$ by utilizing the existing optimized parameters $\theta^*$ obtained through optimizing $L_\phi(\theta)$ without retraining the model. Thus, a disturbance loss function $L_U^\epsilon(\theta)$ is constructed to connect $L_U(\theta)$ to $L_\phi(\theta)$ as follows:

$$L_U^\epsilon(\theta) = \frac{|S|}{|S| - |U|} L_\phi(\theta) + \epsilon \sum_{z\in U} L(z,\theta), \tag{8}$$

where $\epsilon$ is a disturbance factor. As $\epsilon$ tends to $-\frac{1}{|S|-|U|}$, $L_U^\epsilon(\theta)$ tends to $L_U(\theta)$. As $\epsilon$ tends to 0, $L_U^\epsilon(\theta)$ tends to $\frac{|S|}{|S|-|U|}L_\phi(\theta)$. Suppose the optimal parameters of the two loss functions are as follows:

$$\theta_u^\epsilon = argmin(L_U^\epsilon(\theta)), \tag{9}$$

$$\theta^* = argmin(L_\phi(\theta)). \tag{10}$$

Then, we have the following conclusion:

$$\nabla L_\phi(\theta^*) = 0, \tag{11}$$

$$\nabla L_u^\epsilon(\theta_u^\epsilon) = 0. \tag{12}$$

Because the first derivative of the loss function at the optimal parameter is 0 according to the basic theory of machine learning. Put $\theta_u^\epsilon$ into Eq. (8), and take the derivative of both sides with respect to $\theta_u^\epsilon$, then we can get:

**Table 1** Symbol definitions

| Symbol | Definition |
|---|---|
| S | Total related training samples |
| U | A counterfactual group consisting of training samples |
| $\theta$ | Parameters of the recommendation model |
| $L(z,\theta)$ | Loss on training sample $z$ |
| $L_\phi(\theta)$ | Original overall loss |
| $L_U(\theta)$ | Overall loss when remove $U$ from training samples |
| $L_U^\epsilon(\theta)$ | A disturbance loss function to connect $L_U(\theta)$ to $L_\phi(\theta)$ |
| $\epsilon$ | A disturbance factor |
| $\theta_u^\epsilon$ | The optimal parameters of $L_U^\epsilon(\theta)$ |
| $\theta^*$ | The optimal parameters of $L_\phi(\theta)$ |
| $\nabla$ | First derivative |
| $H_{\theta^*}$ | Hessian matrix about $\theta^*$ |
| $\hat{y}_{u,i}$ | Predicted rating given to item i by user u |
| $\hat{y}_{u,i}^{-U}$ | Predicted rating given to item i by user u when remove $U$ from training samples |
| $S_u$ | The set of interaction pairs of user $u$ |
| $S_{rec}$ | The set of interaction pairs containing original recommended item $rec$ |

$$0 = \nabla L_u^\epsilon(\theta_u^\epsilon) = \frac{|S|}{|S|-|U|}\nabla L_\phi(\theta_u^\epsilon) + \epsilon \sum_{z\in U} L(z,\theta_u^\epsilon). \tag{13}$$

For simplicity, let $p = \frac{|S|}{|S|-|U|}$ in the following derivation. If the gradient function $\nabla L_\phi(\theta_u^\epsilon)$ and the loss function $L(z,\theta_u^\epsilon)$ do a second-order Taylor expansion at $\theta^*$, the following equation can be obtained:

$$\begin{aligned} 0 = {} & p\nabla L_\phi(\theta^*) + p\nabla^2 L_\phi(\theta^*)(\theta_u^\epsilon - \theta^*) \\ & + \epsilon \sum_{z\in U}[\nabla L(z,\theta^*) + \nabla^2 L(z,\theta^*)(\theta_u^\epsilon - \theta^*)]. \end{aligned} \tag{14}$$

According to Eq. (11), the above expression can be simplified as follows:

$$[p\nabla^2 L_\phi(\theta^*) + \epsilon \sum_{z\in U} \nabla^2 L(z,\theta^*)](\theta_u^\epsilon - \theta^*) = -\epsilon \sum_{z\in U} \nabla L(z,\theta^*). \tag{15}$$

Individual influence function [22] considers the case of removing one training sample, i.e., $|U| = 1$. They believe that the size of overall samples is large enough (i.e., $|S|$ is large enough) that when $|U| = 1$, the coefficient $\epsilon \to -\frac{1}{|S|-|U|}$ is too small compared to the coefficient $p = \frac{|S|}{|S|-|U|}$, so the part $\epsilon \sum_{z\in U} \nabla^2 L(z,\theta^*)$ on the left side of the equation can be ignored. Then, putting $\nabla^2 L_\phi(\theta^*)$ at the right side of the equation and take the derivative of $\epsilon$ on the both sides as follows:

$$\frac{d(\theta_u^\epsilon - \theta^*)}{d\epsilon} = -H_{\theta^*}^{-1} \sum_{z\in U} \nabla L(z,\theta^*), \tag{16}$$

where $H_{\theta^*} = \nabla^2 L_\phi(\theta^*)$ is the Hessian matrix of the original loss $L_\phi(\theta)$. The parameter $\theta^*$ on the left side of the equation can be omitted because $\theta^*$ is completely independent of $\epsilon$. When $|U| = 1$, we have the same conclusion with Eq. (2). Then, the derivation of the change in parameters is extended to the change in the recommendation results. Consequently, the influence function is obtained as follows:

$$\begin{aligned} \hat{y}_{u,i} - \hat{y}_{u,i}^{-U} &\approx \frac{d\hat{y}_{u,i}}{d\epsilon}\Delta\epsilon \\ &= \frac{1}{|S|-|U|}\frac{d\hat{y}_{u,i}}{d\theta_u^\epsilon}\frac{d\theta_u^\epsilon}{d\epsilon} \\ &= \frac{1}{|S|-|U|}\nabla_\theta \hat{y}_{u,i}(\theta^*) H_{\theta^*}^{-1} \sum_{z\in U} \nabla_\theta L(z,\theta^*), \end{aligned} \tag{17}$$

where $\epsilon = \frac{1}{|S|-|U|}$ because as $\epsilon$ changes from 0 to $-\frac{1}{|S|-|U|}$, the optional parameter changes from $\theta^*$ to $\theta_U$ and the output changes from $\hat{y}_{u,i}$ to $\hat{y}_{u,i}^{-U}$. However, it is not all training samples that affect the result in the field of recommendation, but a few training samples related to the current user and the recommended items [24], which means that $|S|$ is not extremely large. As shown in Fig. 2, for a specific recommendation result (the question mark), only the interaction records of the same users and the collaborative filtering information of the recommended items may affect it during the training pro-

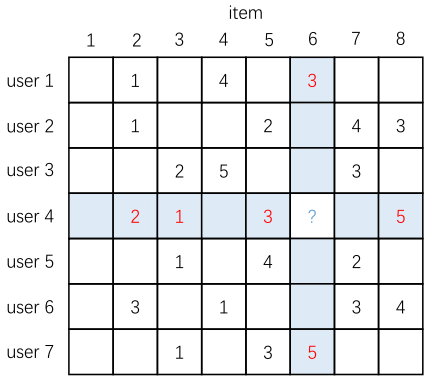


**Fig. 2** A simple illustration for candidate training samples for recommendation

cess (the light blue background). Specifically, there are only 6 records in total (red font).

Thus, when we want to find a counterfactual group for a specific recommendation, there may be a situation where the total number of available training samples is small ($|S|$ is small), in which case omitting $\epsilon \sum_{z\in U} \nabla^2 L(z, \theta^*)$ on the left of Eq. (13) would result in large bias, because the ratio $p/\epsilon = -|S|$ of the two coefficients of $p\nabla^2 L_\phi(\theta^*)$ and $\epsilon \sum_{z\in U} \nabla^2 L(z, \theta^*)$ is not so extremely large that $\epsilon \sum_{z\in U} \nabla^2 L(z, \theta^*)$ can be ignored. Furthermore, if we use the simple summation of individual influence function described in Eq. (2) to express the group influence, the above bias is then accumulated, leading to less accurate estimates of the group influence as the group gets larger.

However, it is almost impossible to give an analytical solution to $\frac{d\hat{\theta}_\epsilon}{d\epsilon}$ according to Eq. (13) without removing $\epsilon \sum_{z\in U} \nabla^2 L(z, \theta^*)$ from the left side. Reference [41] employ perturbation theory [42] to develop second-order influence functions for quantifying the influence of specific groups on model outcomes. Their approach necessitates extensive computational resources due to the requirement of multiple second derivative calculations and the Hessian matrix. Moreover, their methodology exclusively focuses on assessing the influence of known deterministic groups on the model, making it unavailable for searching counterfactual groups.

## Counterfactual group searching method based on modified group influence function

In this subsection, we propose a counterfactual group searching method that addresses the limitation of ACCENT [25]. Unlike ACCENT which calculates group influence by simply adding individual influence functions, we define and utilize a modified group influence function as a basis for searching counterfactual groups.

First, the modified group influence function is defined as follows:

$$\begin{aligned} M\_IN(U, \hat{y}_{u,i}) &\stackrel{def}{=} \hat{y}_{u,i} - \hat{y}_{u,i}^{-U} \\ &= -\frac{1}{|S| - |U|} \sum_{z_t \in U} \nabla_\theta \hat{y}_{u,i}(\theta^*) \widetilde{H}_{\theta^*}^{-1} \nabla_\theta L(z_t, \theta^*), \end{aligned} \tag{18}$$

where $U$ is the counterfactual group. We name $\widetilde{H}_{\theta^*}$ as modified hessian matrix and it is defined as follows:

$$\widetilde{H}_{\theta^*} = \nabla^2 L_\phi(\theta^*) - \frac{1}{|S|} \sum_{z\in U} \nabla^2 L(z, \theta^*), \tag{19}$$

As we argue that ignoring $\epsilon \sum_{z\in U} \nabla^2 L(z, \theta^*)$ from Eq. (13) to deduce Eq. (2) may lead to inaccuracies in subsequent calculations, we take $\sum_{z\in U} \nabla^2 L(z, \theta^*)$ in proportion to the coefficients and we modify $H_{\theta^*}$ to get $\widetilde{H}_{\theta^*}$.

The target of our proposed counterfactual group searching method is to find a group of training samples from the relevant training sample set such that removing this group and retraining the model results in a higher score for the candidate recommendation item $rec^*$ than the original recommendation item $rec$ from related. In other words, the task of counterfactual explanation can be simplified as follows:

$$\begin{aligned} &\text{Find} \quad U \in S_u \cup S_{rec}, \\ &\text{s.t.} \quad \hat{y}_{u,rec}^{-U} < \hat{y}_{u,rec^*}^{-U}, \end{aligned}$$

where $S_u$ is the set of interaction pairs of user $u$ and $S_{rec}$ is the set of interaction pairs containing original recommended item $rec$, and $S = S_u \cup |U|$ represents the related training samples of the current interaction pair.

Although it is not possible for any optimization method to instantly identify counterfactual groups. The strategy we adopt is to continuously select the available training sample that has the greatest potential to change the recommendation results to add counterfactual groups, and in the process, we constantly adjust $H_{\theta^*}$ to ensure accuracy. First, we define the counterfactual influence function score of a training point $z_t$ to judge its potential to change the recommendation results as follows:

$$\begin{aligned} CIF(z_t) &= M\_IN(z_t, \hat{y}_{u,rec}) - M\_IN(z_t, \hat{y}_{u,rec^*}) \\ &= -\frac{1}{|S| - |U|} (\nabla_\theta \hat{y}_{u,rec}(\theta^*) \\ &\quad - \nabla_\theta \hat{y}_{u,rec^*}(\theta^*)) \widetilde{H}_{\theta^*}^{-1} \nabla_\theta L(z_t, \theta^*), \end{aligned} \tag{20}$$

where $\hat{y}_{u,rec}$ is the prediction score of the original recommended item for user $u$ while $\hat{y}_{u,rec^*}$ is the prediction score of the candidate item. $M\_IN(z_t, \hat{y}_{u,rec})$ is a modified individual influence function, defined similarly to Eq. (18). The only difference is that $z_t$ is not included in the counterfactual set $U$

when calculating $\widetilde{H}_{\theta^*}^{-1}$. Specifically speaking, $\widetilde{H}_{\theta^*}$ used here is stored from the previous calculation after selecting $z_{t-1}$. Because calculating the inverse of hessian matrix requires extremely large computing resources, even computing $\widetilde{H}_{\theta^*}$ needs $O((|S| - |U|)k^2)$ operation and reverting $\widetilde{H}_{\theta^*}$ needs $O(k^3)$ operation, where $k$ is the number of parameters. Thus, if we calculate a corresponding $\widetilde{H}_{\theta^*}$ for every training sample $z$ in the counterfactual candidate set $S/U$ when selecting $z$, we need $O((|S| - |U|)((|S| - |U|)k^2 + k^3))$ operations, which leads to losing the simplicity of the influence function.

We employ an iterative approach to extract samples with the highest counterfactual influence function scores from the relevant training samples and add them to the counterfactual group. This process continues until the recommendation results change. The specific algorithm is shown in Algorithm 1.

**Algorithm 1** MGIF

**Input:** user $u$; the recommendation item $rec$; vicarious item $rec^*$,
training samples related to the current recommendation
$S = S_u \cup S_{rec}$. Where $S_u$ is the set of interaction pairs of
user $u$ and $S_{rec}$ is the set of interaction pairs containing item $rec$.
**Output:** counterfactual set $U$.
Initialize the counterfactual group set $U$ as empty, Initialize gap as
$\hat{y}_{rec} - \hat{y}_{rec^*}$
**while** $gap > 0$ **do**
  **for** $z \in S \backslash U$ **do**
    compute $M\_IN(z, \hat{y}_{u,rec})$ and $M\_IN(z, \hat{y}_{u,rec^*})$
    $CIF(z_t) = M\_IN(z_t, \hat{y}_{u,rec}) - M\_IN(z_t, \hat{y}_{u,rec^*})$
  **end for**
  select $z_t$ with the largest $CIF(z_t)$
  **if** $CIF(z_t) > 0$ **then** $z_t$ join the counterfactual group $I_u^*$
    $gap = (\hat{y}_{rec} - M\_IN(U, \hat{y}_{u,rec})) - (\hat{y}_{rec^*} - M\_IN$
$(U, \hat{y}_{u,rec^*}))$
    update $\widetilde{H}_{\theta^*}$
  **else** break
  **end if**
  **if** iterations exceed the maximum number of times **then** break
  **end if**
**end while**
**if** $\hat{y}_{rec}^{-U} < \hat{y}_{rec^*}^{-U}$ **then** Announce success in finding the
counterfactual group
**end if**

An important point to note is that in practical calculations, when it comes to inverting the Hessian matrix, we do not directly compute $\widetilde{H}_{\theta^*}$. Instead, we employ a more efficient second-order optimization method [24] to calculate the Hessian-vector product, thereby avoiding the complex computations involved in directly inverting the Hessian matrix.

Our method cannot directly determine if a recommendation has a counterfactual explanation and precisely find its analytical solution. In reality, this is nearly impossible. Therefore, we adopted a method of step-by-step exploration and continuously adjusting parameters to approach the optimal solution. Empirically, this approach proves to be more accurate than simply using the sum of individual influence functions to represent the group influence function, advancing the field of counterfactual explanations further.

## Experiments

In this section, we present experimental evaluations to demonstrate the superior performance of our proposed MGIF in discovering counterfactual explanations. To provide clear guidance for our experimental analyses, we formulate four research questions (RQ) as follows:

- **RQ1** Does our MGIF framework outperform other explanation methods?
- **RQ2** How do collaborative filtering information contribute to the performance? Can it help improve the performance of MGIF?
- **RQ3** How does MGIF perform groups with different sparsity levels?
- **RQ4** Are the counterfactual predictions given by our MGIF method accurate or not?

### Setup and evaluation protocol

We apply the proposed explanation method MGIF and several competitive baselines on two classic recommendation models, i.e., Matrix Factorization (MF) [43] and Neural Collaborative Filtering(NCF) [44], and evaluate the performance of these explanation methods on two public recommendation datasets, i.e., ML100k and filmtrust. All experiments are done using a single RTX3090 GPU with pytorch framework.

**Datasets.** The ML-100k dataset [45] is the smallest one in the MovieLens series of datasets. It has been sourced from the "MovieLens" website by GroupLens Research. Following the preprocessing steps outlined in [25], the ML-100k dataset has been refined, resulting in a reduced size of 452 users and 682 movies. On the other hand, the FilmTrust dataset [46] is also a small dataset extracted from the comprehensive FilmTrust website. It comprises 13,221 rating records contributed by 660 users, covering 760 movies. We intentionally select these two small datasets for our experiments as the model necessitates retraining on each sample for verification purposes. Estimating the computational resources and time required for counterfactual verification on larger datasets presents considerable challenges.

**Baselines.** To verify the effectiveness of our proposed MGIF method, we select several recently competitive explanation methods as the baselines.

- Pure FIA [24] solely considers the influence of a training sample on the target recommended item $rec$, ignoring its influence on the vicarious item $rec^*$. In other

words, Pure FIA simply sorts the training samples by $M\_IN(z_t, \hat{y}_{u,rec})$ and adds samples to the counterfactual group one by one until $rec$ is displaced. This approach represents the simplest and most straightforward application of the influence function.

- FIA [24] retains only those training samples that effectively minimize the score gap between and $rec$ and $rec^*$. By doing so, FIA further narrows the search for suitable counterfactual groups.
- ACCENT [25] sorts the training samples according to the decreasing $I(z_k, \hat{y}_{rec} - \hat{y}_{rec^*})$ and adds them one by one to the counterfactual groups utile the cumulative sum of $I(z_k, \hat{y}_{rec} - \hat{y}_{rec^*})$ is reached. As we criticize, ACCENT assumes that the summation of the individual influence functions can adequately represent the influence exerted by the group.

**Evaluation Metrics.** For each user $u$, a recommendation model will provide him/her with a top-K recommendation list (we only consider the case where k = 5). Among the recommendations, the topmost item is referred to as the recommended item, $rec$, while the remaining items in the list are considered as vicarious items, denoted as $rec^*$. To assess the effectiveness of explanation methods, we employ a counterfactual approach. Specifically, for each pair $(rec, rec^*)$, we utilize the explanation method to identify a counterfactual group, denoted as $I_u$, which serves as an explanation for the recommendation. Subsequently, we remove the counterfactual group $I_u$ from the dataset and retrain the recommendation model. Finally, we use the retrained model to generate a new recommendation list for the user $u$. To determine the performance of the counterfactual group, we compare the ranking of the vicarious item $rec^*$ with respect to the recommended item $rec$ in the new recommendation list. If $rec^*$ is ranked ahead of $rec$, we define the counterfactual group as precise, assigning a counterfactual precision (CF precision) value of 1 for the $(rec, rec^*)$ pair; otherwise, the CF precision value is set to 0. Additionally, if $rec^*$ is ranked at the top of the new recommendation list, we consider the counterfactual group as effective, assigning a counterfactual effectiveness (CF effectiveness) value of 1.

## Overall performance

To answer **RQ1** and **RQ2**, we summarize the overall performance of our proposed MGIF (with and without collaborative filtering information) as well as the selected baselines in terms of CF precision, CF effectiveness and CF set size in Table 2.

Overall, the proposed MGIF method consistently demonstrates the state-of-the-art performance across all metrics and datasets. Specifically, as shown in Table 2, we observe that

**Table 2** Performance comparison between MGIF and the baselines, where bold values are the best of each column and $*$ denotes statistical significance over ACCENT on t test

| Dataset | | ML-100k | | | Filmtrust | | |
|---|---|---|---|---|---|---|---|
| RS model | Explanation method | CF precision (%) | CF effectiveness (%) | CF set size | CF percentage (%) | CF effectiveness (%) | CF set size |
| MF | Pure FIA | 40.55 | 31.75 | 6.75 | 49.77 | 38.43 | 6.25 |
| | FIA | 50.17 | 42.33 | 4.95 | 52.76 | 45.38 | 4.79 |
| | ACCENT | 68.77 | 53.21 | 2.95 | 72.83 | 54.88 | 2.88 |
| | MGIF | 72.15 | 59.88* | 2.35 | 76.32 | 60.15* | 2.45 |
| | MGIF (with cf) | **78.41*** | **65.37*** | **1.85*** | **82.85*** | **63.77*** | **1.76*** |
| NCF | Pure FIA | 36.08 | 26.86 | 9.23 | 37.77 | 25.24 | 8.95 |
| | FIA | 43.96 | 33.09 | 8.13 | 40.14 | 29.88 | 8.15 |
| | ACCENT | 45.18 | 36.80 | 4.95 | 52.55 | 41.37 | 3.87 |
| | MGIF | 52.62* | 44.02* | 4.73 | 57.37* | 48.85* | 3.55 |
| | MGIF(with cf) | **56.03*** | **46.95*** | **3.97*** | **61.25*** | **52.25*** | **2.75*** |

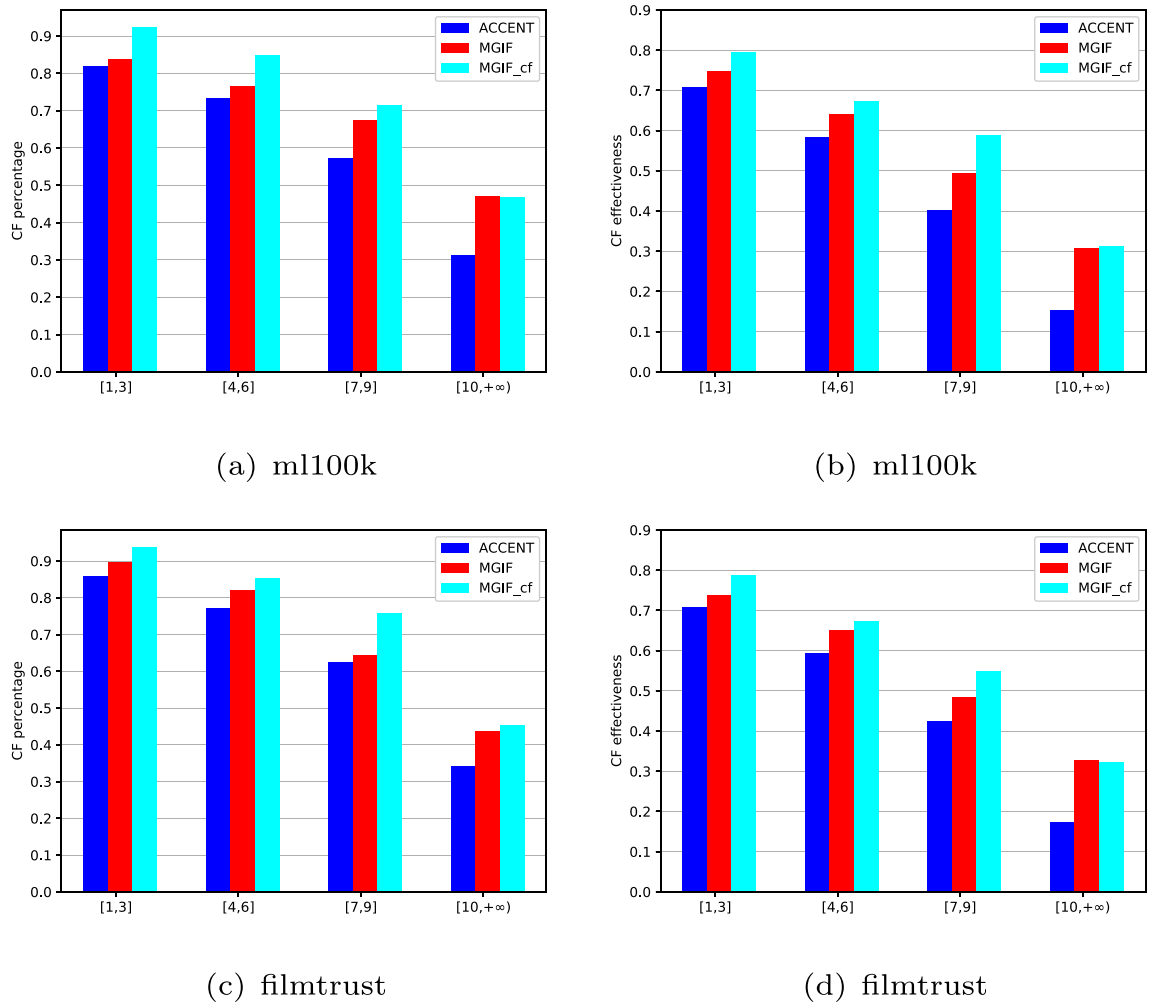


**Fig. 3** Model's Explanation performance on different sparsity groups

MGIF outperforms all other methods in terms of finding accurate counterfactual explanations, regardless of whether MF or NCF is used as the recommendation model. Notably, even without the aid of collaborative filtering information, MGIF exhibits superior performance compared to all other methods, particularly in terms of CF effectiveness where it significantly outperforms ACCENT for both datasets and models. This suggests that our iterative approach for identifying counterfactual groups is highly effective in promoting alternative items to the top of a new recommendation list.

In addition, we note that the performance of various explanation methods on NCF is comparatively worse than that on MF, which is likely due to the increased complexity and number of parameters in NCF, making it more challenging to predict counterfactuals. Additionally, we find no significant difference in counterfactual group size between MGIF and ACCENT, as both methods rely solely on the user interaction history as candidate sets.

However, when collaborative filtering information is incorporated into our proposed method (denoted as MGIF with cf), we observe a significant improvement in performance across three metrics, as well as a notable reduction in the size of counterfactual explanations. These results highlight the importance of collaborative filtering information in generating accurate and effective counterfactual explanations, and demonstrate its significant impact on the recommendation results.

### Performance on groups with different sparsity levels.

To further examine the performance of the models across varying lengths of the counterfactual group (referred to as **RQ3**), the counterfactual group is sorted in ascending order based on its length, defined as the number of training samples, and then evenly divided into four subgroups. ACCENT, MGIF, and MGIF (with cf) are exclusively tested with MF in this particular subsection. The group performance of these methods is illustrated in Fig. 3.

The findings depicted in Fig. 3 reveal a discernible trend, wherein a smaller counterfactual group size corresponds to improved performance of each method across the two indicators. This observation underscores the complexity of the influence of numerous samples on recommendation outcomes, indicating that higher sample volumes result in less accurate predictions. Consequently, the utilization of a simplistic monomer influence function for predicting group influence on results is deemed inappropriate. Importantly, despite all counterfactual methods exhibiting inferior performance on larger groups compared to smaller groups, our proposed MGIF and MGIF (with cf) demonstrate significant superiority over ACCENT on large groups, particularly with respect to CF, where MGIF and MGIF (with cf) exhibit nearly double the performance of ACCENT. This substantiates the appropriateness of our proposal for seeking counterfactual explanations.

Additionally, an intriguing phenomenon is observed within the largest group (comprising more than 10 samples), wherein collaborative filtering information does not appear to enhance the performance of MGIF. This suggests that the superiority of our proposal is not solely attributable to the inclusion of collaborative filtering information. In essence, the improved performance of MGIF (with cf) compared to MGIF is attributed to the collaborative filtering's role in shortening counterfactual groups, thereby enhancing the performance of counterfactual interpretation methods on shorter groups.

## Performance of counterfactual prediction

In the counterfactual explanation task, the explanatory methods aim to find a counterfactual group that can change the target item score $y_{u,rec}$ and candidate item scores $y_{u,rec^*}$. In this process, the method's ability to predict the scores after removing the counterfactual group is crucial. In other words, whether a method can accurately predict the values of $\hat{y}_{u,rec}$ and $\hat{y}_{u,rec^*}$ after removing a training sample group largely determines its ability to accurately find the counterfactual group. Therefore, in this section, we will compare the counterfactual score prediction ability of the explanation methods.

To facilitate the presentation and answer **RQ4**, we randomly select 200 samples and use the predicted values $\hat{y}_{u,rec}$ or $\hat{y}_{u,rec^*}$ as the x-axis, while the actual values after removing the corresponding counterfactual group are plotted as the y-axis. We only present the results on the ML100K dataset because similar phenomenon is observed on the filmtrust dataset.

Based on Fig. 4a, b, d and e, we can observe that the scores predicted by MGIF are more correlated with the ground truth compared to ACCENT. This is an important factor contributing to MGIF's ability to better identify counterfactual groups. From Fig. 4c and f, we can see that the predictions of MGIF (with-cf) are slightly better than MGIF, but the improvement is not significant. This may indicate that the inclusion of collaborative filtering information is most important in providing more sample choices to reduce the length of counterfactual groups, rather than improving the precision of the search. Since all models generally perform better when counterfactual groups are smaller, MGIF (with-cf) has the best performance in terms of counterfactual explanations.

Comparing the first and second rows of the images, it is evident that predictions based on matrix factorization (MF) are consistently more accurate than those based on neural collaborative filtering (NCF). This observation further highlights the greater difficulty in predicting counterfactuals for complex models. Additionally, when comparing Fig. 4a and b, we can observe that the predicted scores for candidate items are generally higher than those for target items (scatter points leaning towards the top right corner). This pattern indicates that counterfactual predictions are mostly successful in the context of MF. However, this phenomenon is not as pronounced in Fig. 4d and e.

## Case study of counterfactual explanation

In order to facilitate a more intuitive comparison between the counterfactual explanations derived from our proposed method and those generated by the ACCENT algorithm, we randomly select two illustrative cases from the experimental results, as detailed in Table 3. The analysis reveals that our proposed MGIF method yields more succinct counterfactual explanations in contrast to ACCENT. This outcome can be attributed to the enhancements made to the group influence function, which augment the model's capacity to accurately identify counterfactual groups, as evidenced by the counterfactual explanation of User 2. Furthermore, the integration of collaborative filtering information enriches the counterfactual samples and broadens the search scope for counterfactual groups, thereby increasing the probability of identifying pivotal and decisive counterfactual samples, as observed in the counterfactual explanation of User 9.

Specifically, the examination of the counterfactual explanation of User 2 reveals that ACCENT and MGIF typically identify the same counterfactual samples in the initial step, unless they originate from different user interactions. This can be attributed to MGIF not adjusting the Hessian matrix (H) in the first step, essentially conducting the same computation as ACCENT. Furthermore, the analysis of the counterfactual explanation of User 9 demonstrates that collaborative filtering interactions from different users can also serve as crucial components of the counterfactual explanation, as exemplified by User 288's interaction with "Full Metal Jacket" identified in the first step of MGIF. Nevertheless, the semantic relationship between the counterfactual samples and the recommended items from the samples

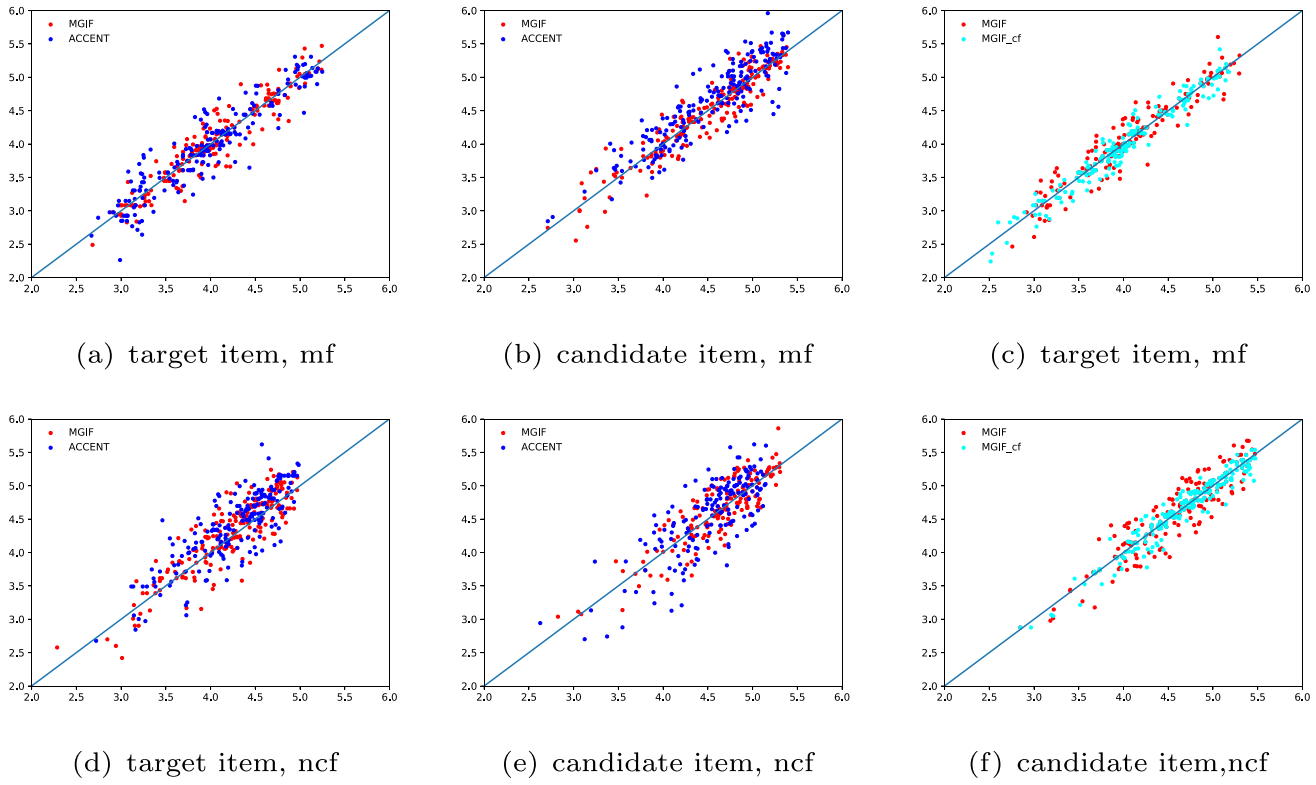


**Fig. 4** Counterfactual score prediction on ml100k. x-axis represents the recommendation rating predicted by counterfactual method and y-axis means the true rating after removing the corresponding counterfactual group. The top row represents the results of recommendation models based on matrix factorization(mf), while the bottom row represents the results of recommendation models based on Neural Collaborative Filtering (ncf). **b** and **e** both show the score predictions for candidate items, while the rest of the images display the score predictions for target items. The results for ACCENT, MGIF, and MGIF (with cf) are represented in blue, red, and cyan colors, respectively. The blue line highlights the $y = x$ line, which represents the ground truth

**Table 3** Counterfactual explanation sets generated by MGIF and ACCENT methods

| User | Recommendation | Counterfactual explanation | | Replacement |
|---|---|---|---|---|
| User 2 (53, female, other) | "Godfather,II" | ACCENT | You liked "Crow" | "Thinner" |
| | | | You liked "I.Q." | |
| | | | You liked "Toy Story" | |
| | | MGIF | You liked "Crow" | |
| | | | You liked "Godfather" | |
| User 9 (29, male, student) | "Full Metal Jacket" | ACCENT | You liked "His Girl Friday" | "Aliens" |
| | | | You liked "Fierce Creatures" | |
| | | | You liked "Sleepers" | |
| | | MGIF | User 288 (34, M, marketing) liked "Full Metal Jacket" | |
| | | | You liked "His Girl Friday" | |

themselves is not clearly discernible. This may be due to the model's reliance solely on interaction information for learning, without the availability of auxiliary information, thereby limiting its capacity to explicitly learn associations from user-item interaction behavior. Notably, we find that the observed cosine similarity between the embeddings of "Crow" and "Godfather, II" is 0.86, underscoring the complexity of providing persuasive explanations for recommendations exclusively based on interaction information in the absence of auxiliary data.

## Conclusion

In this paper, we propose a method for explaining recommendations based on an improved group influence function. Firstly, we demonstrate, through formula derivation, that the utilization of a simple summation of individual influence functions to represent the group influence function introduces a significant bias. Subsequently, we construct counterfactual groups by sequentially incorporating samples from the training set, while continuously adjusting the Hessian matrix $H$ to maximize accuracy. Moreover, we expand the scope of searching for counterfactual groups by incorporating collaborative filtering information from different users. We conduct extensive experiments on two publicly available datasets to showcase the efficacy of our proposed model and affirm that collaborative filtering information aids in identifying more concise and accurate counterfactual groups.

The challenge of explaining recommendations without relying on auxiliary information remains a significant topic of interest. In future work, we aim to further validate our proposed method on larger datasets and deploy it in real-world applications to evaluate user satisfaction with the explanations.

**Funding** ZK22-11, Scientific Research Project of National University of Defense Technology.



## References

1. Yang L, Wang S, Tao Y, Sun J, Liu X, Yu P.S, Wang T (2023) Dgrec: graph neural network for recommendation with diversified embedding generation. In: Proceedings of the sixteenth ACM international conference on web search and data mining. WSDM 2023, Singapore, 27 February 2023–3 March 2023, pp 661–669
2. Qin Y, Wang Y, Sun F, Ju W, Hou X, Wang Z, Cheng J, Lei J, Zhang M (2023) DisenPOI: disentangling Sequential and Geographical Influence for Point-of-Interest Recommendation. In: Proceedings of the sixteenth ACM international conference on Web search and data mining. WSDM 2023, Singapore, 27 February 2023–3 March 2023, pp 508–516
3. Qin J, Zhu J, Liu Y, Gao J, Ying J, Liu C, Wang D, Feng J, Deng C, Wang X, Jiang J, Liu C, Yu Y, Zeng H, Zhang W (2023) Learning to distinguish multi-user coupling behaviors for TV recommendation. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023–3 March 2023, pp 204–212
4. Quan Y, Ding J, Gao C, Yi L, Jin D, Li Y (2023) Robust preference-guided denoising for graph based social recommendation. In: Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023, pp 1097–1108
5. Chen J, Song L, Wainwright M.J, Jordan MI (2018) Learning to explain: an information-theoretic perspective on model interpretation. In: Proceedings of the 35th international conference on machine learning. ICML 2018, Stockholmsmässan, Stockholm, July 10–15, 2018, pp 882–891
6. Chen G, Chen J, Feng F, Zhou S, He X (2023) Unbiased knowledge distillation for recommendation. In: Proceedings of the sixteenth ACM international conference on web search and data mining, WSDM 2023, Singapore, 27 February 2023-3 March 2023, pp 976–984
7. Wang Z, Zhu Y, Wang C, Ma W, Li B, Yu J (2023) Adaptive graph representation learning for next POI recommendation. In: ACM, SIGIR conference on research and development in information retrieval. SIGIR, pp 393–402
8. Li R, Zhang L, Liu G, Wu J (2023) Next basket recommendation with intent-aware hypergraph adversarial network. In: SIGIR, pp 1303–1312
9. Zhang J, Chen X, Tang J, Shao W, Dai Q, Dong Z, Zhang R (2023) Recommendation with causality enhanced natural language explanations. In: Proceedings of the ACM web conference 2023. WWW 2023, Austin, TX, USA, 30 April 2023–4 May 2023, pp 876–886
10. Du Y, Lian J, Yao J, Wang X, Wu M, Chen L, Gao Y, Xie X (2023) Towards explainable collaborative filtering with taste clusters learning. In: Proceedings of the ACM web conference 2023. WWW 2023, Austin, TX, USA, 30 April 2023–4 May 2023, pp 3712–3722
11. Ribeiro MT, Singh S, Guestrin C (2016) "Why should I trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. San Francisco, CA, USA, August 13–17, 2016, pp 1135–1144
12. Balog K, Radlinski F (2020) Measuring recommendation explanation quality: the conflicting goals of explanations. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. SIGIR 2020, Virtual Event, China, July 25–30, 2020, pp 329–338
13. Ghazimatin A, Pramanik S, Roy R.S, Weikum G (2021) ELIXIR: learning from user feedback on explanations to improve recommender models. In: WWW '21: the web conference 2021. Virtual event/Ljubljana, Slovenia, April 19–23, 2021, pp 3850–3860
14. Xian Y, Zhao T, Li J, Chan J, Kan A, Ma J, Dong XL, Faloutsos C, Karypis G, Muthukrishnan S (2021) EX3: explainable attribute-

aware item-set recommendations. In: RecSys '21: fifteenth ACM conference on recommender systems, Amsterdam, The Netherlands, 27 September 2021-1 October 2021 , pp 484–494
15. Zhang W, Yan J, Wang Z, Wang J (2022) Neuro-symbolic interpretable collaborative filtering for attribute-based recommendation. In: proceedings of the ACM web conference 2022. pp 3229–3238
16. Markchom T, Liang H, Ferryman J (2023) Scalable and explainable visually-aware recommender systems. Knowl Based Syst 263:110258
17. Cai Z, Cai Z (2022) Pevae: A hierarchical vae for personalized explainable recommendation. In: SIGIR '22: the 45th international ACM SIGIR conference on research and development in information retrieval, Madrid, Spain, July 11–15 pp. 692–702
18. Park S-J, Chae D-K, Bae H-K, Park S, Kim S-W (2022) Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable recommendation. In: WSDM '22: the fifteenth ACM international conference on web search and data mining, virtual event / Tempe, AZ, USA, February 21–25, 2022. pp 784–793
19. Wang X, Wang D, Xu C, He X, Cao Y, Chua T-S (2019) Explainable Reasoning over Knowledge Graphs for Recommendation. In: The Thirty-Third AAAI onference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27-February 1, 2019, pp 5329–5336,
20. Kgtn: Knowledge graph transformer network for explainable multi-category item recommendation. Knowl Based Syst 278: 110854 (2023)
21. Wiegreffe S, Pinter Y (2019) Attention is not not Explanation. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019 pp 11–20
22. Koh PW, Liang P (2017) Understanding black-box predictions via influence functions. In: Proceedings of the 34th international conference on machine learning. Proceedings of machine learning research, vol 70. ICML 2017, Sydney, NSW, Australia, 6–11 August 2017. 70: 1885–1894
23. Basu S, You X, Feizi S (2020) On second-order group influence functions for black-box predictions. In: Proceedings of the 37th international conference on machine learning, ICML 2020, 13–18 July 2020, Virtual Event. Proceedings of machine learning research, pp 715–724
24. Cheng W, Shen Y, Huang L, Zhu Y (2019) Incorporating interpretability into latent factor models via fast influence analysis. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, KDD 2019, Anchorage, AK, USA, August 4–8, pp 885–893
25. Tran K, Ghazimatin A, Roy RS (2021) Counterfactual explanations for neural recommenders. In: SIGIR '21: the 44th international ACM SIGIR conference on research and development in information retrieval, virtual event, Canada, July 11–15, pp1627–1631
26. Todorovic M, Stanisic N, Zivkovic M, Bacanin N, Simic V, Tirkolaee EB (2023) Improving audit opinion prediction accuracy using metaheuristics-tuned xgboost algorithm with interpretable results through shap value analysis. Appl Soft Comput 149:110955
27. Alabi RO, Elmusrati M, Leivo I, Almangush A, Mäkitie AA (2023) Machine learning explainability in nasopharyngeal cancer survival using lime and shap. Sci Rep 13(1):8984
28. Abdollahi A, Pradhan B (2023) Explainable artificial intelligence (xai) for interpreting the contributing factors feed into the wildfire susceptibility prediction model. Sci Total Environ 879:163004
29. Hada DV, Shevade SK (2021) Rexplug: explainable recommendation using plug-and-play language model. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp 81–91
30. Li L, Zhang Y, Chen L (2023) Personalized prompt learning for explainable recommendation. ACM Trans Inf Syst 41(4):1–26
31. Ren G, Diao L, Guo F, Hong T (2024) A co-attention based multi-modal fusion network for review helpfulness prediction. Inf Process Manag 61(1):103573
32. Chen X, Zhang Y, Qin Z (2019) Dynamic explainable recommendation based on neural attentive models. In: proceedings of the AAAI conference on artificial intelligence 33(1):53–60
33. Li L, Zhang Y, Chen L (2021) Personalized Transformer for Explainable Recommendation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing ACL/IJCNLP 2021. Volume 1: Long Papers), Virtual Event, August 1–6, 2021, Association for Computational Linguistics, pp 4947–4957
34. Cheng Z, Ding Y, Zhu L, Kankanhalli M (2018) Aspect-aware latent factor model: Rating prediction with ratings and reviews. In: Proceedings of the 2018 world wide web conference on world wide web, WWW 2018, Lyon, France, April 23–27. pp 639–648
35. Fu Z, Xian Y, Gao R, Zhao J, Huang Q, Ge Y, Xu S, Geng S, Shah C, Zhang Y (2020) Fairness-aware explainable recommendation over knowledge graphs. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, pp 69–78
36. Wei T, Chow TW, Ma J, Zhao M (2023) Expgcn: review-aware graph convolution network for explainable recommendation. Neural Netw 157:202–215
37. Gao J, Peng P, Lu F, Claramunt C, Xu Y (2023) Towards travel recommendation interpretability: Disentangling tourist decision-making process via knowledge graph. Inf Process Manag 60(4):103369
38. Chen H, Shi S, Li Y, Zhang Y (2021) Neural collaborative reasoning. In: Proceedings of the web conference 2021. pp 1516–1527
39. Shi S, Chen H, Ma W, Mao J, Zhang M, Zhang Y (2020) Neural logic reasoning. In: Proceedings of the 29th ACM international conference on information and knowledge management. pp 1365–1374
40. Xu Z, Zeng H, Tan J, Fu Z, Zhang Y, Ai Q (2023) A reusable model-agnostic framework for faithfully explainable recommendation and system scrutability. ACM Trans Inf Syst 42:1–29
41. Basu S, You X, Feizi S (2020) On second-order group influence functions for black-box predictions. In: International conference on machine learning. PMLR, pp 715–724
42. Brueckner K (1968) Perturbation theory and its applications. In: Mathematical methods in solid state and superfluid theory: scottish iniversities' summer school. pp 235–285
43. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer, IEEE 42(8):30–37
44. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. In:Proceedings of the 26th international conference on world wide web, WWW 2017, Perth, Australia, April 3–7. pp 173–182
45. Tan EYH (2023) A critical study on movielens dataset for recommender systems
46. Guo G, Zhang J, Yorke-Smith N (2013) A novel bayesian similarity measure for recommender systems. In: Proceedings of the 23rd international joint conference on artificial intelligence (IJCAI). pp 2619–2625