A Causal View for Item-level Effect of Recommendation on User Preference

Wei Cai
cai_wei@zju.edu.cn
Zhejiang University
Hangzhou, China

Fuli Feng*
fulifeng93@gmail.com
University of Science and Technology of China, Hefei, China

Qifan Wang
wqfcr@fb.com
Meta AI
Menlo Park, United States

Tian Yang
tyang@cuhk.edu.hk
Chinese University of Hong Kong
Hong Kong, China

Zhenguang Liu
liuzhenguang2008@gmail.com
Zhejiang University
Hangzhou, China

Congfu Xu*
xucongfu@zju.edu.cn
Zhejiang University
Hangzhou, China

ABSTRACT

Recommender systems not only serve users but also affect user preferences through personalized recommendations. Recent researches investigate the effects of the entire recommender system on user preferences, *i.e.,* system-level effects, and find that recommendations may lead to problems such as echo chambers and filter bubbles. To properly alleviate the problems, it is necessary to estimate the effects of recommending a specific item on user preferences, *i.e.,* item-level effects. For example, by understanding whether recommending an item aggravates echo chambers, we can better decide whether to recommend it or not.

This work designs a method to estimate the item-level effects from the causal perspective. We resort to causal graphs to characterize the average treatment effect of recommending an item on the preference of another item. The key to estimating the effects lies in mitigating the confounding bias of time and user features without the costly randomized control trials. Towards the goal, we estimate the causal effects from historical observations through a method with stratification and matching to address the two confounders, respectively. Nevertheless, directly implementing stratification and matching is intractable, which requires high computational cost due to the large sample size. We thus propose efficient approximations of stratification and matching to reduce the computation complexity. Extensive experimental results on two real-world datasets validate the effectiveness and efficiency of our method. We also show a simple example of using the item-level effects to provide insights for mitigating echo chambers.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

item-level effects, recommendation effects, causal inference, stratification, matching

ACM Reference Format:
Wei Cai, Fuli Feng, Qifan Wang, Tian Yang, Zhenguang Liu, and Congfu Xu. 2023. A Causal View for Item-level Effect of Recommendation on User Preference. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570461

1 INTRODUCTION

Recommender system is a medium to connect users and items, which is a cornerstone of various online platforms [4, 14, 35]. It aims to facilitate information seeking by recommending items that match user preferences. There is a growing consensus that recommender systems also affect user preferences while catering to the user preferences [4, 14, 25]. Recent researches focus on the effect of the entire recommender system on user preferences, *i.e.,* the system-level effects. Specifically, they attempt to answer whether and to what extent a system changes the preference of users. For example, some exceptional studies [3, 25, 36] reveal that recommender systems in scenarios such as e-commerce narrow and reinforce users' interests, leading to notorious issues of echo chambers and filter bubbles. Other studies [14, 18] measure the degree of the echo chambers caused by the systems.

The recognition of system-level effects calls for the consideration of such effects when making recommendation. Nevertheless, the current understanding of system effects is too coarse to guide the specific adjustment of recommendations. To better determine whether to recommend an item, we need to quantify the specific effect of recommending this item on user preferences, *i.e.,* item-level effect. For example, if we find that recommending an item contributes to the exacerbation of echo chambers, we can reduce the frequency of recommending this item. Nevertheless, few studies focus on estimating the item-level effects. Therefore, we aim to design an effective method to quantify such effect.

Given that interactions with items reveal user preferences, we set the target as answering a causal question: *to what extent recommending an item to a user affects the probability of like on another item.* To recognize the causal effect, we abstract a causal graph to describe the generation of interactions. As shown in Figure 1, we assume whether the user likes on item j (L) at time (T) is affected

*Co-corresponding authors. This paper is supported by the National Key R&D Program of China under grant 2022YFB3104700, and partially supported by the National Natural Science Foundation of China (NSFC) under grant 61672449 and 62272437.

WSDM '23, February 27-March 3, 2023, Singapore, Singapore
© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570461

by the previous exposure of item v (R). User features (F) simultaneously affect the exposure probability of item v and like probability of item j. Accordingly, our target is to estimate the average treatment effect (ATE) [15, 57] of R on L for each item pairs (v, j).

A default choice for estimating ATE is randomized control trials [12, 28], which require two groups of users, one as the treatment group and the other as the control group. Each user in the treatment group is forcibly recommended item v. After that, we can estimate ATE as the difference of like probabilities over the two groups when recommending item j. However, randomized controlled trials are time-consuming and may harm user experience [23]. Therefore, we estimate ATE from observational records of exposure and likes. The core idea is selecting two user groups such that their historical records can be regarded as-if they are treated by randomized control trails. The key lies in conducting the confounder adjustment to make the exposure of item v and confounders independent. A widely used confounder adjustment is stratification [22, 57], which is employed to eliminate the time confounder. Nevertheless, stratification fails to deal with high-dimensional confounders, in this case the user feature confounder. In this light, we leverage matching [40] to deal with the confounder of user features.

Estimating the ATE for all (v, j) item pairs is challenging due to the large size of items[1]. The computational cost of existing stratification and matching methods [22, 39] (*e.g.,* propensity scores [32]) is unaffordable. We thus pursue efficient approximations of stratification and matching. For stratification, we establish the independence between exposure and confounders with an ingenious sampling of users that largely reduces the computation cost. As to matching, we propose a heuristic distance metric that significantly accelerates the matching process with binary search. We conduct an extensive set of experiments on two public real-world datasets. (i) Experimental results show that our method successfully eliminates both sets of confounders, validating the effectiveness of our proposed method and approximation strategies. (ii) Experiments demonstrate that our method is significantly more efficient than the existing methods that use the propensity scores. (iii) We show a simple example to illustrate how to use the item-level effects to guide the design of recommender systems.

The main contributions of this work are as follows:

- We study a new problem of estimating the item-level effects on user preferences and solve the problem through confounder adjustment methods with stratification and matching.
- We propose two efficient approximation strategies of stratification and matching, which largely reduce the time complexity of estimating the item-level effects.
- We conduct sufficient experiments that demonstrate the effectiveness and efficiency of our method, and show the significance of the item-level effects on the design of recommender systems.

2 PROPOSED METHOD

Problem Definition. For each item pair (v, j), our target is to estimate the ATE [15, 57] of recommending the item v on the user preferences over item j from the relevant historical records. Let $\mathcal{X} = \{x_i\} = \{(u_i, t_i, f_i, r_i, l_i)\}$ denote the relevant historical records, where i is the index of samples. A sample $x = (u, t, f, r, l)$ denotes the feedback l of user u with features f on the recommendation of item j at time t. l is a binary indicator of whether the user u likes[2] the item j, where $l = 1$ if u likes j at t and $l = 0$ otherwise. Similarly, r denotes the exposure of item v to user u at time t, where $r = 1$ if u has been recommended item v before t and $r = 0$ otherwise. In the following, we first scrutinize the problem from the causal perspective in Section 2.1. Then we introduce the elimination of two confounders in Section 2.2 and 2.3 respectively.

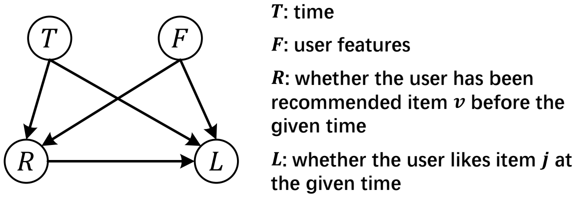

T
F
R
L
T: time
F: user features
R: whether the user has been recommended item *v* before the given time
L: whether the user likes item *j* at the given time

Figure 1: Causal relations of recommending an item v (R) and user preference on item j (L). T and F are confounders between R and L.

2.1 Recognizing Recommendation Effects

To recognize the effects, we abstract the generation procedure of historical records in Figure 1. In the causal graph, we use T, F, R, L to denote the relevant variables: time, user feature, exposure of item v, and feedback on item j, respectively. A directed edge indicates that the value of the successor node depends on the value of the ancestor node. In particular,

- $T \longrightarrow R$ indicates that time affects the probability of recommending item v. Such effects can be strong in certain scenarios such as news recommendation and micro-video recommendation.
- $T \longrightarrow L$ indicates the probability of interacting with item j changes over time. For example, the interaction probability will decrease as news gets out-of-date.
- $F \longrightarrow R$ indicates user features influence the probability of recommending item v. This is because recommender models typically consider user features when making recommendations.
- $F \longrightarrow L$ indicates the user features affect the probability of interacting with item j. For instance, the gender of the user directly affects whether an item will be purchased.

Average Treatment Effect. According to the causal graph, our target is to estimate the ATE over path $R \longrightarrow L$, *i.e.,* from the treatment (recommending item v) to the outcome (liking item j). Conceptually, given an item pair (v, j) and the relevant records $\mathcal{X} = \{(u_i, t_i, f_i, r_i, l_i)\}$, ATE represents the difference of the probabilities of liking j between samples in the treatment group ($\{x_i \mid r_i = 1\}$) and control group ($\{x_i \mid r_i = 0\}$) [19]. In an ideal case, we can collect the samples through randomized controlled trials, and directly estimate the ATE by computing the difference over the two groups, which is formulated as:

$$\mathrm{ATE}^{\mathrm{ideal}} = \overline{L}^{\mathrm{t}} - \overline{L}^{\mathrm{c}} = \frac{\sum_i r_i l_i}{\sum_i r_i} - \frac{\sum_i (1 - r_i) l_i}{\sum_i (1 - r_i)}, \tag{1}$$

where $\overline{L}^{\mathrm{t}}$ and $\overline{L}^{\mathrm{c}}$ denote the average L values of the treatment group and the control group, respectively.

[1] Every pair requires one pass of stratification and matching.

[2] In practice, we can consider any kind of user feedback (*e.g.,* "click") as the "like", which indicates the preference of users.

Estimating from History. In practice, it is improper to conduct this randomized controlled trial-based estimation in recommendation, since the exposure policy under control can hurt user experience. Moreover, each item pair requires a set of trials, which is unaffordable given the large number of items. Therefore, we estimate the ATE from the historical records $\mathcal{X}$ collected from previous recommendations. As aforementioned, calculating the ATE from historical records faces confounding biases due to the presence of the confounders T and F [15, 22]. A default choice for eliminating the confounding biases is performing confounder adjustment [22, 57] over the historical records. These adjustment methods select treatment and control groups from $\mathcal{X}$ such that the confounder and treatment are independent over the selected samples, *i.e.,* removing the effect of confounder on the treatment. In the following sections, we describe how to estimate the ATE for (v, j) item pairs effectively and efficiently with the confounder adjustment.

2.2 Adjusting for Time Confounder

Following previous causal studies [5, 45], we choose the widely used stratification [22, 57] to eliminate the time confounder T. We first introduce the key concepts in stratification methods (Section 2.2.1) and the implementation of stratification with propensity scores (Section 2.2.2). Then we propose an approximation strategy to reduce the computation cost (Section 2.2.3). Lastly, the time complexity analysis is provided (Section 2.2.4).

2.2.1 Stratification Methods. To eliminate the time confounder T, we need to cut off the backdoor path $T \rightarrow R$, or make T and R independent of each other. The stratification method [22, 57] is a typical approach to adjust confounders. The idea is to split the data into multiple subgroups such that the confounder T and the treatment R are independent within each subgroup.

Formally, given the data $\mathcal{X} = \{x_i\}$, we first divide $\mathcal{X}$ into K disjoint subgroups $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_K$, assuming that T and R are independent for the samples in each subgroup (details will be provided later), *i.e., $T \perp\!\!\!\perp R$*. Since the time confounder is eliminated, we can accurately estimate the ATE in each subgroup:

$$\text{ATE}^{\text{s}}_k = \bar{L}^{\text{t}}_k - \bar{L}^{\text{c}}_k = \frac{\sum_{x_i \in \mathcal{X}_k} r_i l_i}{\sum_{x_i \in \mathcal{X}_k} r_i} - \frac{\sum_{x_i \in \mathcal{X}_k} (1 - r_i) l_i}{\sum_{x_i \in \mathcal{X}_k} (1 - r_i)}. \tag{2}$$

The results of each subgroup are then combined to estimate ATE:

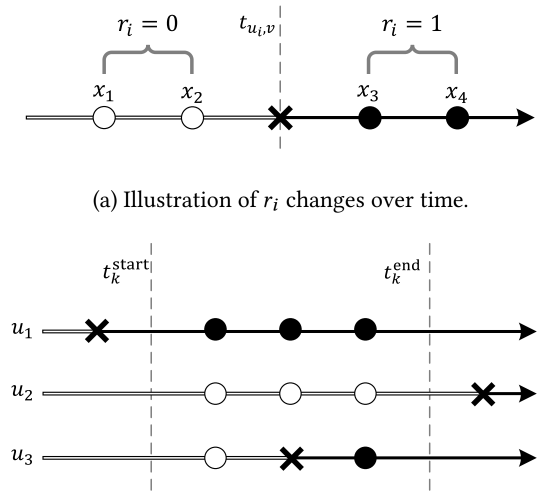
$$\text{ATE} = \sum_{k=1}^{K} q_k \text{ATE}^{\text{s}}_k, \; q_k = \frac{|\mathcal{X}_k|}{|\mathcal{X}|}, \tag{3}$$

where q_k denotes the proportion of the k-th subgroup.

2.2.2 Stratification with Propensity Scores. One of the key problems in stratification methods is how to divide the data to ensure the samples within the same subgroup satisfying $T \perp\!\!\!\perp R$. A common approach is to stratify based on the propensity scores [32, 39]. The propensity scores (PS) are defined as the conditional probabilities of the treatment given the confounder. Specifically, for (v, j) item pair, the propensity score is defined as $P(R = 1|v, T = t_i)$. We put the samples with the same propensity scores into a subgroup:

$$\mathcal{X}^{\text{ps}}_k = \{x_i \mid P(R = 1|v, T = t_i) = p_k\}, \tag{4}$$

where p_k is the common PS of the samples in the k-th subgroup. Since the conditional probabilities are fixed, $T \perp\!\!\!\perp R$ is satisfied.



(a) Illustration of r_i changes over time.

(b) llustration of three types of users.

Figure 2: Illustration of users' historical data. Arrows: The timelines. Fork: One recommendation for the item v. Circle: One recommendation for the item j (*i.e.,* a sample in our data), where white indicates $r = 0$ and black indicates $r = 1$.

However, there are two problems when calculating the propensity scores from historical data. First, it is difficult to estimate propensity scores accurately [41], and the inaccurate estimation could introduce additional challenges to our stratification. Second, it is impractical to estimate the effects over all (v, j) item pairs regardless of calculating the propensity scores as pre-processing or on the fly. The pre-calculation of the propensity scores is unrealistic due to the storage cost to cover all combinations of v and t. As such, calculating the propensity scores when estimating the effect also face an unaffordable time cost.

2.2.3 Stratification with Approximation Strategy. To resolve the above issues, we propose an approximation strategy. The idea is to efficiently establish the independence of T and R in each subgroup by applying an ingenious user sampling strategy, hence eliminating the time confounder with much less computational cost. Recall that r_i indicates whether the user u_i has been recommended with the item v before time t_i. As shown in Figure 2 (a), assuming that user u_i is first recommended with item v at time $t_{u_i,v}$, we have:

$$r_i = \begin{cases} 1, & t_i > t_{u_i,v} \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

which means that the $t_{u_i,v}$ is the cut-off point for the value of R. We treat samples over a continuous period of time as a subgroup:

$$\mathcal{X}_k = \left\{x_i \mid t_k^{\text{start}} \le t_i < t_k^{\text{end}}\right\}, \tag{6}$$

where the start time t_k^{start} and end time t_k^{end} for each subgroup can be set by controlling the group size $|\mathcal{X}_k|$. More details are provided in the experiments. Then within each subgroup, the users can be naturally classified into three categories:

- Users who have been recommended v before t_k^{start}, *i.e.,* $\mathcal{U}_1 = \{u_i \mid t_{u_i,v} < t_k^{\text{start}}\}$ (see u_1 in Figure 2 (b)). In this case, we have:

$$P(R=1|T=t, u_i \in \mathcal{U}_1) = 1 = P(R=1|u_i \in \mathcal{U}_1). \tag{7}$$

In other words, R is always equal to 1 regardless of the value of T, since the user has been recommended v before the subgroup start time t_k^{start}. Therefore, T and R are independent in this case.

- Users who are first recommended v after t_k^{end} (u_2 in Figure 2 (b)), *i.e.,* $\mathcal{U}_2 = \{u_i \mid t_{u_i,v} \geq t_k^{\text{end}}\}$, there is:

$$P(R=1|T=t, u_i \in \mathcal{U}_2) = 0 = P(R=1|u_i \in \mathcal{U}_2). \tag{8}$$

Similarly, R is always equal to 0 and we have $T \perp\!\!\!\perp R$ in this case.

- The third category of users is defined as $\mathcal{U}_3 = \{u_i \mid t_k^{\text{start}} < t_{u_i,v} \leq t_k^{\text{end}}\}$, where u_3 in Figure 2 (b) is an example in this category. Since r_i may vary with t_i, we usually have $T \not\perp\!\!\!\perp R$ for $u_i \in \mathcal{U}_3$. However, the number of samples in this case is negligible compared to those in the first two cases, especially for a small time interval. Therefore, we approximate the estimation by ignoring the samples in the last case:

$$\begin{aligned} \mathcal{X}'_k &= \mathcal{X} \backslash \{x_i \mid x_i \in \mathcal{X}_k, u_i \in \mathcal{U}_3\} \\ &= \{x_i \mid x_i \in \mathcal{X}_k, u_i \in \mathcal{U}_1 \cup \mathcal{U}_2\}. \end{aligned} \tag{9}$$

Then for samples in the set $\mathcal{X}'_k$, there is:

$$\begin{aligned} &P(R=1|T=t) \\ \overset{(a)}{=} &\sum_{u \in \mathcal{U}_1 \cup \mathcal{U}_2} P(R=1|U=u, T=t) P(U=u|T=t) \\ \overset{(b)}{=} &\sum_{u \in \mathcal{U}_1 \cup \mathcal{U}_2} P(R=1|U=u) P(U=u|T=t) \\ \overset{(c)}{=} &\sum_{u \in \mathcal{U}_1 \cup \mathcal{U}_2} P(R=1|U=u) P(U=u) \overset{(d)}{=} P(R=1), \end{aligned} \tag{10}$$

where Eq. (10)(a) is the Law of total probability; Eq. (10)(b) is the conditional independence mentioned in Eq. (7) and Eq. (8); Eq. (10)(c) is from the basic independence assumption of U and T. As Eq. (10)(d) showed, the samples in the subgroup $\mathcal{X}'_k$ satisfy $T \perp\!\!\!\perp R$. In summary, we can eliminate the time confounder by simply removing a small fraction of the samples.

2.2.4 Time Complexity. With the approximation strategy, dividing subgroups and removing samples only require $O(N \log K)$, where N is the number of samples and K is the number of subgroups. The time complexity of the estimation is linear $O(N)$. Therefore, the total time complexity is $O(N \log K + N) = O(N \log K)$. Without the approximation strategy, the propensity scores need to be computed first. The total time complexity is $O(NS + N \log K)$, where S is the size of the item representations used to calculate the propensity scores. In practice, S is much larger than $\log K$. Therefore, our approximation strategy dramatically reduces the time cost.

2.3 Adjusting for User Feature Confounder

After the stratification, the time confounder is decoupled from the ATE estimation. However, the samples in each subgroup are still affected by the user feature confounder, where stratification is not suitable in this case as it fails to deal with the high-dimensional confounders. Therefore, we employ the matching methods [40] to eliminate the user feature confounder in each subgroup. This section first introduces the concept of the matching methods (Section 2.3.1). We then propose several effective distance metrics for matching (Section 2.3.2 - 2.3.4) and analyze the total computational costs (Section 2.3.5).

2.3.1 Matching Methods. We denote $l_i(r_i = 1)$ as the value of l_i assuming $r_i = 1$. When the i-th sample belongs to the treatment group ($r_i = 1$), $l_i(r_i = 1)$ is in fact happened and thus is called the factual outcome. In this case, there is $l_i(r_i = 1) = l_i$. When the i-th sample belongs to the control group ($r_i = 0$), $l_i(r_i = 1)$ is also exist, it just does not happen in fact. Therefore, when $r_i = 0$, $l_i(r_i = 1)$ is called the counterfactual outcome. The difference between the factual outcome and counterfactual outcome is the ATE.

Matching methods [40] are a representative category of methods for estimating the counterfactual outcome. In particular, they match similar sample in another group based on some distance metrics, and use the outcome of the matched sample to estimate the counterfactual outcome.

Formally, $l_i(r_i = 1)$ can be estimated as:

$$\hat{l}_i(r_i = 1) = \begin{cases} l_i, & r_i = 1 \\ l_{i'}, & r_i = 0, \end{cases} \tag{11}$$

where $\hat{l}_i(r_i = 1)$ is the estimated value of $l_i(r_i = 1)$ and i' is the index of the matched sample. Similarly, we estimate $l_i(r_i = 0)$ as:

$$\hat{l}_i(r_i = 0) = \begin{cases} l_{i'}, & r_i = 1 \\ l_i, & r_i = 0. \end{cases} \tag{12}$$

After estimating the factual and counterfactual outcomes, we can directly calculate the ATE as:

$$\text{ATE}_k = \sum_{x_i \in \mathcal{X}'_k} \left(\hat{l}_i(r_i = 1) - \hat{l}_i(r_i = 0) \right). \tag{13}$$

Note that ATE_k is not affected by the user feature confounder. So we use ATE_k instead of ATE_k^{S} to compute $\text{ATE} = \sum_{k=1}^{K} q_k \text{ATE}_k$.

2.3.2 Matching with Propensity Scores. For the matching methods, the distance metric used for matching directly affects the accuracy of the estimated counterfactual outcomes. One strategy is to use propensity scores to define the distance metric [1]. The principle of this metric is similar to the principle of stratification with propensity scores in Section 2.2.2. Concretely, for the i-th sample of (v, j) item pair, we first define the propensity score as $P(R = 1|v, F = f_i)$. Then, the distance metric is defined as:

$$d_{i,i'}^{\text{ps}} = |P(R=1|v, F=f_i) - P(R=1|v, F=f_{i'})|, \tag{14}$$

where $d_{i,i'}$ denotes the distance between the i-th and i'-th samples.

Similar to the drawbacks of propensity scores discussed in stratification, it is also challenging to estimate the propensity scores in matching. First, the estimated propensity scores have high variances [41], which generate unstable matching results. Second, the huge number of (v, f) values requires us to compute many propensity scores, leading to high computational costs.

2.3.3 Matching with Approximation Strategy. In order to reduce the computational cost, we propose a heuristic distance metric to accelerate the matching with binary search. Specifically, we design a new heuristic distance metric based on two important user features.

First, as the treatment group consists of users that have been recommended item v, many selected users have been exposed to abundant recommended items. These users often have a lower like rate, compared to those in the control group, as they are very selective in liking and interacting with the huge amount of items presented to them, resulting in a bias in the estimation (we further investigate this phenomenon in Section 3.2). To address this problem, we define a distance metric such that the users of the matched sample pairs have similar like rates:

$$d^{\mathrm{r}}_{i,i'} = \left|h^{\mathrm{r}}(u_i) - h^{\mathrm{r}}(u_{i'})\right|, \tag{15}$$

where $h^{\mathrm{r}}(u_i)$ represents the historical average like rate of user u_i.

Second, we observe that users in the treatment group prefer the items in the category g_v. Here g_v denotes the item category that the item v belongs to. In other words, if the item j also belongs to g_v, these users are more likely to prefer the item j as well. This bias misleads the estimation of the effect to be large for (v, j) item pair when both v and j come from the same item category (more investigation is discussed in Section 3.2). To eliminate this bias, we design another distance metric such that the users of the matched sample pairs have similar preferences for the item category g_v.

$$d^{\mathrm{vr}}_{i,i'} = \left|h^{\mathrm{vr}}(u_i) - h^{\mathrm{vr}}(u_{i'})\right|, \tag{16}$$

where $h^{\mathrm{vr}}(u_i)$ denotes the historical average like rate of users on the item category g_v. Finally, we combine the two distance metrics:

$$\begin{aligned} d_{i,i'} &= \frac{1}{2}\left(d^{\mathrm{r}}_{i,i'} + d^{\mathrm{vr}}_{i,i'}\right) \\ &= \frac{1}{2}\left[\left|h^{\mathrm{r}}(u_i) - h^{\mathrm{r}}(u_{i'})\right| + \left|h^{\mathrm{vr}}(u_i) - h^{\mathrm{vr}}(u_{i'})\right|\right]. \end{aligned} \tag{17}$$

This metric is able to eliminate both biases mentioned above. To speed up the matching, we modify $d_{i,i'}$ with an approximation:

$$d'_{i,i'} = \frac{1}{2}\left|\left(h^{\mathrm{r}}(u_i) + h^{\mathrm{vr}}(u_i)\right) - \left(h^{\mathrm{r}}(u_{i'}) + h^{\mathrm{vr}}(u_{i'})\right)\right|. \tag{18}$$

Note that $d'_{i,i'} = d_{i,i'}$ when $h^{\mathrm{r}}(u_i) - h^{\mathrm{r}}(u_{i'})$ and $h^{\mathrm{vr}}(u_i) - h^{\mathrm{vr}}(u_{i'})$ have the same sign. The advantage of d' is that it enables fast matching with binary search after pre-calculating the value of $\left(h^{\mathrm{r}}(u_i) + h^{\mathrm{vr}}(u_i)\right)$ for each sample.

2.3.4 Matching with Embeddings. Although matching with metric d' is efficient, it only eliminates the bias of the features that are used in the distance metric. Another alternative is to adopt a comprehensive but inefficient metric using user feature embeddings, *i.e.,* latent user representations:

$$d^{\mathrm{e}}_{i,i'} = \frac{1}{2}\left(1 - \frac{e_{u_i} e_{u_{i'}}}{\|e_{u_i}\|\|e_{u_{i'}}\|}\right), \tag{19}$$

where e_{u_i} denotes the embedding of user u_i, and we follow Mult-VAE [30] to learn the user embeddings.

2.3.5 Time Complexity. With our approximation metric d' and the binary search, the time complexity of matching is $O(N^{\mathrm{t}} \log N^{\mathrm{c}})$, where N^{t} and N^{c} are the number of samples in the treated and control groups, respectively. Without the approximation strategy, the time complexity increases to $O(NS + N^{\mathrm{t}} \log N^{\mathrm{c}})$. $O(NS)$ is the complexity to calculate the propensity scores, where $N = N^{\mathrm{t}} + N^{\mathrm{c}}$ denotes the number of all samples and S denotes the size of the user representations. In practice, $O(NS)$ is the dominant term. Therefore, our approximation metric significantly accelerates the matching.

• **Summary.** We summarize the ATE estimation method with adjustment for the confounders in Algorithm 1. We start by stratifying $\mathcal{X}$ into subgroups. Then we match samples in each subgroup and merge the ATE_k of each subgroup into ATE.

Algorithm 1 The method for estimating the item-level effects.

```
Input: $\mathcal{X} = \{x_i\} = \{(u_i, t_i, f_i, r_i, l_i)\}$ of the item pair $(v, j)$
Output: ATE of the item pair $(v, j)$
// Stratification
Divide $\mathcal{X}$ into $K$ disjoint subgroups $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_K$
for $k = 1$ to $K$ do
    $\mathcal{X}'_k := \mathcal{X}_k \backslash \{x_i \mid x_i \in \mathcal{X}_k, t^{\mathrm{start}}_k < t_{u_i,v} \le t^{\mathrm{end}}_k\}$
end for
// Matching in each subgroup
for $k = 1$ to $K$ do
    for each sample $x_i$ in the treatment group of $\mathcal{X}'_k$ do
        $\hat{l}_i(r_i = 1) := l_i$
        Binary search the matched sample $x_{i'}$ with a minimum
        $d_{i,i'}$ in the control group of $\mathcal{X}'_k$
        $\hat{l}_i(r_i = 0) := l_{i'}$ // $i'$ is the index of the matched sample $x_{i'}$
    end for
    $\mathrm{ATE}_k := \sum_{x_i \in \mathcal{X}'_k} \left(\hat{l}_i(r_i = 1) - \hat{l}_i(r_i = 0)\right)$
end for
// Merge the $\mathrm{ATE}_1, \mathrm{ATE}_2, \ldots, \mathrm{ATE}_K$ into ATE
ATE $:= \sum_{k=1}^{K} q_k \mathrm{ATE}_k$
```

Table 1: Statistics of the datasets after preprocessing.

Statistics	#User	#Item	#Exposure	#Click	#Category
MIND	750,434	29,309	97,592,931	3,958,501	16
CW	30,465	7,261	3,555,038	329,870	4

3 EXPERIMENTS

In this section, we aim to answer the following research questions:

- RQ1: Does our method eliminate the two kinds of confounders?
- RQ2: How is the efficiency of the proposed approximation strategies compared with propensity-based approaches?
- RQ3: How does the item-level effects estimated by our method provide insights for designing better recommender systems?

3.1 Datasets

We use two public real-world datasets from different domains to test our method. (i) **MIND**[3] [55]. The MIND dataset contains records of exposure and click from the Microsoft News[4]. (ii) **ContentWise (CW)**[5] [37], which contains records of exposure and click on video content, such as TV series and movies. Both datasets provide item categories, which are used for the calculation of d^{vr} and d' to perform matching. We treat click as the feedback of user preference (*i.e., L*) for both datasets. Note that we merge the records of exposure and click in the CW dataset according to their timestamps, which are separately provided in the original release. Other than that, we

[3] https://msnews.github.io/

[4] https://microsoftnews.msn.com/

[5] https://github.com/ContentWise/contentwise-impressions

Table 2: Performance of eliminating the time confounder. The best and second best results are marked in bold and underlined, respectively.

Dataset	MIND			CW		
Metric	P<0.01	P<0.05	P<0.1	P<0.01	P<0.05	P<0.1
Direct	40.72%	49.16%	54.84%	59.96%	68.60%	73.28%
RS	26.65%	36.48%	43.37%	52.75%	63.57%	69.30%
Strat	<u>3.79%</u>	<u>8.97%</u>	<u>14.68%</u>	**9.36%**	**18.21%**	**25.54%**
SAM	**3.14%**	**8.76%**	**14.52%**	<u>14.14%</u>	<u>24.43%</u>	<u>32.35%</u>

Table 3: Statistics on user click rates.

Metric	FT	FC	Diff	P<0.01	P<0.05	P<0.1
MIND	0.0345	0.0378	-0.0033	41.28%	50.58%	56.82%
CW	0.0845	0.0876	-0.0031	49.90%	59.80%	65.46%

Table 4: Statistics on user click rates for item category g_v.

Metric	FT	FC	Diff	P<0.01	P<0.05	P<0.1
MIND	0.0367	0.0354	0.0013	28.14%	38.88%	46.00%
CW	0.0833	0.0782	0.0051	48.96%	58.06%	63.82%

keep the original data without further processing. The statistics for the processed datasets are shown in Table 1. In the following experiments, we set the size of subgroup $|\mathcal{X}_k|$ to 1,000 for MIND and 500 for CW since the various scales of two datasets. The source codes are available for reproduction[6].

3.2 Performance Comparison (RQ1)

3.2.1 Performance of Eliminating Time Confounder. We first evaluate whether the proposed stratification method eliminates the time confounder. For each subgroup $\mathcal{X}_k$ of the item pair (v, j), we need to examine whether T in the treatment group ($r_i = 1$) and the control group ($r_i = 0$) are significantly different. We adopt the Welch's t-test [54] to calculate the P value, and denote the P value for subgroup $\mathcal{X}_k$ of the item pair (v, j) as $P_{v,j,k}$. Follow the previous works [20, 26], if $P_{v,j,k}$ is less than some threshold (*e.g.,* 0.01, 0.05, and 0.1), we consider T in the treatment and control groups to be significantly different, which implies that the samples in the subgroup $\mathcal{X}_k$ are still affected by the T confounder. Finally, we calculate the proportion of subgroups affected by the T confounder in all subgroups of all item pairs:

$$\text{P<0.01} = \frac{\sum_{v,j}\sum_k \delta(P_{v,j,k} < 0.01)}{\sum_{v,j}\sum_k 1}, \tag{20}$$

where indicator function $\delta(P_{v,j,k} < 0.01) = 1$ if and only if $P_{v,j,k} < 0.01$ is true. Note that the value of the metric P<0.01 is in the range of $[0, 1]$, and lower value indicates better adjustment for the confounder. Similarly, we calculate P<0.05 and P<0.1 to cover different significant levels, and evaluate the following causal effect estimation methods: (i) **Direct**: The direct estimation without stratification in Eq. (1). (ii) **RS**: Randomly stratify the samples. (iii) **Strat**: Stratification with our proposed approximation strategy as formulated in Eq. (9). (iv) **SAM**: Our method (including both stratification and matching) with approximation strategies.

Table 2 shows the performance where we find that:

Table 5: Performance of eliminating the user click rate bias.

Dataset	MIND			CW		
Metric	P<0.01	P<0.05	P<0.1	P<0.01	P<0.05	P<0.1
Direct	41.28%	50.58%	56.82%	49.90%	59.80%	65.46%
RM	28.32%	39.20%	45.72%	31.92%	43.24%	50.22%
Mat w d^{r}	**0.02%**	**0.02%**	**0.02%**	**0.34%**	**0.36%**	**0.44%**
Mat w d^{vr}	46.98%	55.54%	61.40%	14.30%	22.28%	28.22%
Mat w d^{e}	25.70%	37.26%	44.68%	18.16%	28.14%	35.38%
Mat	16.52%	25.14%	31.04%	8.34%	<u>13.88%</u>	<u>19.22%</u>
SAM	<u>7.92%</u>	<u>18.58%</u>	<u>27.13%</u>	<u>8.04%</u>	14.46%	19.62%

Table 6: Performance of eliminating the bias of user click rates for items in category g_v.

Dataset	MIND			CW		
Metric	P<0.01	P<0.05	P<0.1	P<0.01	P<0.05	P<0.1
Direct	28.14%	38.88%	46.00%	48.96%	58.06%	63.82%
RM	17.38%	26.80%	34.52%	29.84%	41.86%	49.14%
Mat w d^{r}	16.18%	25.84%	32.52%	12.22%	21.74%	28.90%
Mat w d^{vr}	**0.02%**	**0.02%**	**0.02%**	**0.30%**	**0.38%**	**0.44%**
Mat w d^{e}	19.36%	28.58%	35.60%	21.96%	33.16%	40.34%
Mat	4.84%	8.16%	11.50%	3.70%	7.06%	10.82%
SAM	<u>0.77%</u>	<u>3.10%</u>	<u>6.10%</u>	<u>2.88%</u>	<u>6.19%</u>	<u>9.63%</u>

- Compared to Direct and RS, Strat and SAM largely reduce the number of subgroups affected by the time confounder in both datasets. The results suggest that our stratification method and approximation strategy effectively eliminate the time confounder.
- There is a slight difference in the performances of Strat and SAM. We infer that the two confounders might not be completely independent in real datasets. Adjusting the user feature confounder in SAM may also affect the time confounder.

3.2.2 Performance of Eliminating User Feature Confounder. We first justify the conjecture in Section 2.3.3, *i.e.,* user click rates, especially on the items in the same category as item v (*i.e.,* g_v) are sources of estimation bias related to the user feature confounder. We denote the average of user features F in the treatment group and control group as FT and FC, and calculate the difference Diff = FT − FC. Similar to Section 3.2.1, we evaluate the P values for the Welch's t-tests between the feature differences in the treatment group and control group. As shown in Table 3, without matching, the click rates of users in the treatment group are significantly lower than those in the control group, and a large proportion (40%-65%) of the samples are affected by this bias. Similarly, as illustrated in Table 4, users in the treatment group have significantly higher click rates on g_v than those in control group. These results reveal that the user click rates on all items and on items of category g_v are two important user features causing the bias of estimating the ATE.

Next, we analyze whether the proposed matching method eliminates the user feature confounder. In addition to Direct and RM, we further compare our method SAM with: (i) **Mat w d^{r}**, d^{vr} and d^{e}: Matching with our proposed d^{r} (Eq.(15)), d^{vr} (Eq.(16)) and d^{e} (Eq.(19)), respectively. (ii) **Mat**: Matching with the approximation strategy, *i.e.,* mathing with our proposed d' (Eq.(18)). Note that SAM is matched in each subgroup $\mathcal{X}_k$ after stratification, which is consistent with our formulation in Section 2.3. In addition, to individually evaluate whether our matching methods eliminate the

[6] https://github.com/mdyx/Recommendation-Effect

Table 7: Running time with PS and our approximation.

#Sample	1e3	2e3	5e3	1e4	2e4	5e4	1e5
Strat w PS	0.033	0.042	0.093	0.168	0.331	0.791	1.574
Strat	0.010	0.011	0.012	0.014	0.021	0.043	0.087
Mat w PS	0.029	0.043	0.096	0.170	0.335	0.831	1.674
Mat	0.008	0.008	0.011	0.015	0.026	0.060	0.129
SAM w PS	0.059	0.092	0.175	0.346	0.660	1.634	3.242
SAM	0.013	0.014	0.017	0.021	0.035	0.075	0.155

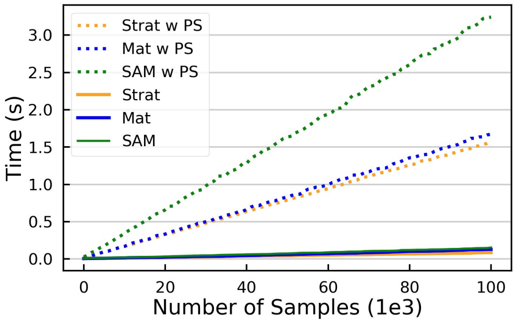


Figure 3: Running time with PS and our approximation.

user feature confounders, RM and Mat w d^{r}, d^{vr}, d^{e}, d' employ matching in the whole group $\mathcal{X}$ rather than the subgroup $\mathcal{X}_k$.

The results of the user click rates on all items and on g_v are presented in Tables 5 and 6, respectively. We observe that:

- Mat w d^{r} and Mat w d^{vr} achieve the best performances in two biases, respectively. The proposed distance metrics d^{r} and d^{vr} alleviate the affects of user feature confounders for almost all samples, illustrating the validity of conducting matching.
- SAM and Mat achieve the second best performance on both datasets, which validates the effectiveness of our distance approximation d'. Moreover, SAM and Mat are effective for both biases, while Mat w d^{r} and Mat w d^{vr} can only eliminate one.
- Mat w d^{e} is not specifically designed for these two biases, thus showing poor performances. But its performance also surpasses that of RM and Direct, validating the rationality of using user embedding for matching.
- SAM shows promising performances in handling both time and user feature confounders, which demonstrates that our entire method successfully mitigates the confounding bias.

3.3 Efficiency Comparison (RQ2)

We then compare the efficiency of implementing the stratification and matching with propensity scores (PS) and the proposed approximation strategies. We record the time to estimate effects at different sample sizes $|\mathcal{X}|$. We omit the results on the CW dataset due to its relatively small size. We compare six approaches:

- Strat w PS, Mat w PS and SAM w PS: Implementing stratification, matching and the entire method (including stratification and matching) with the propensity scores, respectively.
- Strat, Mat and SAM: Implementing stratification, matching and the entire method (including stratification and matching) with our approximation strategies, respectively.

The experimental results are shown in Figure 3. We observe that all the variants of the proposed method are significantly faster than the existing methods with PS. This indicates that our strategies significantly improve the efficiency. To better show the efficiency improvement, we show the quantitative running time in Table 7. As can be seen, the running time of adjustment with PS is 15 to 20 times longer than that with our approximations.

Table 8: Correlation of effects and item similarities.

Pearsonr	Spearmanr	Kendalltau
0.040^{***}	0.036^{***}	0.028^{***}

Table 9: Statistics of ATE$_{v,j}$ for (v, j) pairs that $sim_{v,j}$ >= 0.01.

Positive Ratio	Average	Median
57.0%	0.0075	0.0031

Table 10: Correlation of effects and item similarities in different cases.

Correlation	Pearsonr	Spearmanr	Kendalltau
All	0.040^{***}	0.036^{***}	0.028^{***}
Same Category	0.067^{***}	0.105^{***}	0.079^{***}
Diff Categories	0.015^{*}	0.009	0.008

Table 11: Statistics of ATE$_{v,j}$ for (v, j) pairs that $sim_{v,j}$ >= 0.01 in different cases.

Statistics	Positive Ratio	Average	Median
All	57.0%	0.0075	0.0031
Same Category	65.8%	0.0124	0.0071
Diff Categories	47.7%	0.0024	0.0000

3.4 Application of the Item-level Effects (RQ3)

We then show a simple example of using the item-level ATE estimated by our method to provide insights for the design of recommender systems with the MIND dataset which has recently received more attention [31, 55].

3.4.1 Recognizing Filter Bubbles. Filter bubbles mean that user preferences are constantly reinforced with recommendations [25]. We study whether recommendations in the Microsoft News lead to filter bubbles in this section.

We first investigate the correlation between the effects and item similarities. Specifically, we randomly sample (v, j) item pair and estimate the effect ATE of (v, j), which is denoted as ATE$_{v,j}$. Then we collect users that interact with v and j separately and calculate the Jaccard Index as a proxy gauging the item similarity sim$_{v,j}$. We then calculate correlation coefficients of ATE$_{v,j}$ and sim$_{v,j}$. The results are displayed in Table 8, where we use * to indicate P value < 0.1, ** the 0.05, and *** the 0.01. Although the correlations appear relatively small, all correlations are significant (P value < 0.01). Significant positive correlations between effects and item similarities suggest that the more similar v and j are, the greater the effect of recommendation v on user preferences of j (*i.e.,* making users prefer j).

We then further calculate the ATE$_{v,j}$ of similar item pairs (*e.g.,* $sim_{v,j}$ >= 0.01) and show the statistical properties of ATE in Table 9. We observe that the ratio of ATE$_{v,j}$ $>$ 0 exceeds 50 and both the average and median of ATE$_{v,j}$ are $>$ 0. Note that ATE$_{v,j}$ $>$ 0 means that recommending item v leads the user to prefer item j. Therefore, if a user prefers item j and the system recommends an item v that is similar to the item j, then the recommendation tends to reinforce the user preferences. Considering that a core principle of the recommender systems is to recommend items similar to those prefered by the user, recommendations in the Microsoft News unavoidably lead to filter bubbles.

3.4.2 Alleviating Filter Bubbles. In this section, we further explore the findings in Section 3.4.1. Specifically, we analyse the correlation between the effects and item similarities in following ranges: (i) **All**: All item pairs (same as Section 3.4.1). (ii) **Same Category**: Pairs that v and j are from the same category. (iii) **Diff Categories**: Pairs that v and j are from different categories. We show the results in Table 10 and find that the correlations in Same Category are more significant. Conversely, effects and item similarities are almost completely unrelated in Diff Categories.

We adopt the same approach to analyse the effects of similar item pairs ($sim_{v,j} >= 0.01$) in Table 11. We observe that for user who prefers an item j, recommending an item that is similar to j and belongs to the same category as j is likely (with 65.8% probability) to reinforce the user preferences of the item j. However, recommending an item that is similar to j but belongs to a different category from j is less likely (with 47.7% probability) to reinforce the user preferences of j. This gives a simple and feasible way to alleviate filter bubbles: recommend items that match the user preferences but belong to categories that the user has less interaction with. We believe that more precise ways to leverage the item-level effects could be found in future work.

4 RELATED WORK

In this section, we first discuss studies which focused on related research questions, including the system-level effects on user preferences and immediate effects on user feedbacks. Then we review relevant existing algorithms, especially those that apply the causal inference in the recommender systems.

System-level Effect on User Preference. Most prior studies investigate whether and to what extent the whole recommender system influence user preferences. They roughly fall into two categories. The first line of work focuses on the echo chambers [3] and the filter bubbles [36]. These two refer to that the users with similar interests are aggregated by the recommender systems [48]. Their interests and perspectives are constantly reinforced and amplified in this environment [36, 47]. Some researchers [24, 25] attempt to propose theoretical models to explain these phenomena. They [25] theoretically demonstrate that without intervention, user preferences will be amplified once they interact with the recommender systems. Other researchers, through the analysis of real-world data, find that the echo chambers and the filter bubbles widely exist in many recommendation scenarios such as video media [18, 35], social network [4, 7] and e-commerce [14]. They also propose metrics [14, 18] to demonstrate the extent of the echo chambers. The second line of work investigates the "nudge" of recommender systems on user preferences. These works [8, 11, 27, 34, 50] argue that recommender systems intentionally or unintentionally nudge the preferences in a particular direction. For example, researchers [8, 9] suggest that some recommender systems get users addicted to certain content by recommending specific items (*e.g.,* negative news). However, these works only study the effects of the entire recommender system. The difference is that our method can quantify the specific effects of recommending an item and help us understand whether recommending the item leads to problems such as echo chambers. Therefore, our method can provide specific guidance for the recommendation strategy.

Immediate Effect on User Feedback. Common recommendation algorithms tend to recommend items that will be interacted with (e.g. purchase and click). However, some items could have been interacted with even without recommendation. Therefore, some recent works [12, 13, 28, 56] aim to estimate the immediate effect of recommendation on user feedback about this recommendation. Earlier methods [6, 42] first estimate the interaction probabilities with and without recommendations, and then estimate the effects by looking at the differences between them. Upon this, some methods [13, 43] eliminates bias and provides more accurate recommendations. The main differences between these works and ours are that: (i) they focus on the immediate impact of recommendations on user feedback, while we focus primarily on the long-term impact of recommendations on user preferences, and (ii) they focus on the user feedback on the exposed item, while we aim to estimate the effect of exposure of one item on user preference for another.

Causal Inference in Recommendation. The causal inference focuses on how to eliminate confounding bias [16]. Inverse of propensity scores (IPS) [21, 38], stratification [22, 57], and matching [40] are common causal inference methods. Causal inference has been recently introduced into recommender systems to eliminate various biases [10, 51, 52] such as popularity bias[2], clickbait bias [52] and Matthew effect [51]. Early studies [44, 49] propose unbiased metrics and unbiased learning methods through the IPS methods. Several researchers [29] present exposure models for estimating propensity scores. To improve the robustness of the methods, the doubly robust models [53] combine the IPS methods and the data imputation methods [17, 33]. Very few studies adopt stratification methods and matching methods in recommender systems. Some research [5, 45, 46] stratifies items according to their popularity and design fairer evaluations of recommender systems. In addition, one study [20] employs matching methods to discuss whether recommendations segregate people into different groups. The differences between these algorithms and ours are that: (i) our method is not allowed to use the propensity scores due to the huge computational cost and high variance [41, 53], and (ii) they only eliminate one type of confounders, which is relatively simple.

5 CONCLUSIONS AND FUTURE WORK

This paper highlights the importance of estimating the effects of recommending an item on user preferences, *i.e.,* item-level effects. We adopt the widely used stratification and matching to eliminate the confounding bias. In order to improve the efficiency of the estimation, we present two new approximation strategies. Extensive experiments on two real-world datasets demonstrate that our methods effectively eliminate the bias and that our approximation strategies significantly shorten the running time.

In the future, we would like to investigate whether there exists other biases (*e.g.,* selection bias) in the estimation procedure that are not considered, and enhance our method to eliminate such bias. Moreover, the processing of each sample in our method is fully parallelizable. Therefore, we plan to implement a parallel version of the method to further speed up the estimation.

REFERENCES

[1] Alberto Abadie and Guido W Imbens. 2016. Matching on the estimated propensity score. *Econometrica* 84, 2 (2016), 781–807.

[2] Himan Abdollahpouri and Masoud Mansoury. 2020. Multi-sided exposure bias in recommendation. *arXiv preprint arXiv:2006.15772* (2020).

[3] David Paul Allen, Henry Jacob Wheeler-Mackta, and Jeremy R Campo. 2017. The effects of music recommendation engines on the filter bubble phenomenon. *Interactive Qualifying Projects* (2017).

[4] Eytan Bakshy, Solomon Messing, and Lada A Adamic. 2015. Exposure to ideologically diverse news and opinion on Facebook. *Science* 348, 6239 (2015), 1130–1132.

[5] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2017. Statistical biases in Information Retrieval metrics for recommender systems. *Information Retrieval Journal* 20, 6 (2017), 606–634.

[6] Anand V Bodapati. 2008. Recommendation systems with purchase data. *Journal of marketing research* 45, 1 (2008), 77–93.

[7] Laura Burbach, Patrick Halbach, Martina Ziefle, and André Calero Valdez. 2019. Bubble trouble: strategies against filter bubbles in online social networks. In *International Conference on Human-Computer Interaction*. 441–456.

[8] Christopher Burr, Nello Cristianini, and James Ladyman. 2018. An analysis of the interaction between intelligent software agents and human users. *Minds and machines* 28, 4 (2018), 735–774.

[9] Guillaume Chaslot. 2018. How algorithms can learn to discredit the media. *Medium. Retrieved from https://medium. com/@ guillaumechaslot/how-algorithms-can-learn-to-discredit-the-media-d1360157c4fa* (2018).

[10] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240* (2020).

[11] Katja De Vries. 2010. Identity, profiling algorithms and a world of ambient intelligence. *Ethics and information technology* 12, 1 (2010), 71–85.

[12] M Benjamin Dias, Dominique Locher, Ming Li, Wael El-Deredy, and Paulo JG Lisboa. 2008. The value of personalised recommender systems to e-business: a case study. In *Proceedings of the 2008 ACM conference on Recommender systems*. 291–294.

[13] Sihao Ding, Peng Wu, Fuli Feng, Yitong Wang, Xiangnan He, Yong Liao, and Yongdong Zhang. 2022. Addressing unmeasured confounder for recommendation with sensitivity analysis. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 305–315.

[14] Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, and Yongfeng Zhang. 2020. Understanding echo chambers in e-commerce recommender systems. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 2261–2270.

[15] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

[16] Miguel A Hernán and James M Robins. 2010. Causal inference.

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

[18] Martin Hilbert, Saifuddin Ahmed, Jaeho Cho, Billy Liu, and Jonathan Luu. 2018. Communicating with algorithms: A transfer entropy analysis of emotions-based escapes from online echo chambers. *Communication Methods and Measures* 12, 4 (2018), 260–275.

[19] Paul W Holland. 1986. Statistics and causal inference. *Journal of the American statistical Association* 81, 396 (1986), 945–960.

[20] Kartik Hosanagar, Daniel Fleder, Dokyun Lee, and Andreas Buja. 2014. Will the global village fracture into tribes? Recommender systems and their effects on consumer fragmentation. *Management Science* 60, 4 (2014), 805–823.

[21] Guido W Imbens. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics* 86, 1 (2004), 4–29.

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

[23] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 447–455.

[24] Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. 2019. Degenerate feedback loops in recommender systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 383–390.

[25] Dimitris Kalimeris, Smriti Bhagat, Shankar Kalyanaraman, and Udi Weinsberg. 2021. Preference Amplification in Recommender Systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 805–815.

[26] Eunji Kim, Michael E Shepherd, and Joshua D Clinton. 2020. The effect of big-city news on rural America during the COVID-19 pandemic. *Proceedings of the National Academy of Sciences* 117, 36 (2020), 22009–22014.

[27] Ansgar Koene, Elvira Perez, Christopher James Carter, Ramona Statache, Svenja Adolphs, Claire O'Malley, Tom Rodden, and Derek McAuley. 2015. Ethics of personalized information filtering. In *International Conference on Internet Science*. 123–132.

[28] Dokyun Lee and Kartik Hosanagar. 2014. Impact of recommender systems on sales volume and diversity. (2014).

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

[30] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.

[31] Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. KRED: Knowledge-aware document representation for news recommendations. In *Fourteenth ACM Conference on Recommender Systems*. 200–209.

[32] Jason K Luellen, William R Shadish, and MH Clark. 2005. Propensity scores: An introduction and experimental test. *Evaluation Review* 29, 6 (2005), 530–558.

[33] Benjamin Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2012. Collaborative filtering and the missing at random assumption. *arXiv preprint arXiv:1206.5267* (2012).

[34] Silvia Milano, Mariarosaria Taddeo, and Luciano Floridi. 2020. Recommender systems and their ethical challenges. *AI & SOCIETY* 35, 4 (2020), 957–967.

[35] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*. 677–686.

[36] Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you*. Penguin UK.

[37] Fernando B Pérez Maurera, Maurizio Ferrari Dacrema, Lorenzo Saule, Mario Scriminaci, and Paolo Cremonesi. 2020. ContentWise impressions: an industrial dataset with impressions included. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 3093–3100.

[38] Paul R Rosenbaum. 1987. Model-based direct adjustment. *J. Amer. Statist. Assoc.* 82, 398 (1987), 387–394.

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

[40] Donald B Rubin. 1973. Matching to remove bias in observational studies. *Biometrics* (1973), 159–183.

[41] Yuta Saito. 2020. Asymmetric Tri-training for Debiasing Missing-Not-At-Random Explicit Feedback. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 309–318.

[42] Masahiro Sato, Hidetaka Izumo, and Takashi Sonoda. 2016. Modeling Individual Users' Responsiveness to Maximize Recommendation Impact. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. 259–267.

[43] Masahiro Sato, Sho Takemori, Janmajay Singh, and Tomoko Ohkuma. 2020. Unbiased learning for the causal effect of recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 378–387.

[44] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. 1670–1679.

[45] Harald Steck. 2011. Item popularity and recommendation accuracy. In *Proceedings of the fifth ACM conference on Recommender systems*. 125–132.

[46] Harald Steck and Yu Xin. 2010. A Generalized Probabilistic Framework and its Variants for Training Top-k Recommender System.. In *PRSAT@ RecSys*. 35–42.

[47] Cass Sunstein and Cass R Sunstein. 2018. *# Republic*. Princeton university press.

[48] Cass R Sunstein. 2009. *Going to extremes: How like minds unite and divide*. Oxford University Press.

[49] Adith Swaminathan and Thorsten Joachims. 2015. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research* 16, 1 (2015), 1731–1755.

[50] Mariarosaria Taddeo and Luciano Floridi. 2018. How AI can be a force for good. *Science* 361, 6404 (2018), 751–752.

[51] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded recommendation for alleviating bias amplification. In *KDD*. 1717–1725.

[52] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *SIGIR*. 1288–1297.

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

[54] Bernard L Welch. 1938. The significance of the difference between two means when the population variances are unequal. *Biometrika* 29, 3/4 (1938), 350–362.

[55] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3597–3606.

[56] Di Yao, Chang Gong, Lei Zhang, Sheng Chen, and Jingping Bi. 2021. CausalMTA: Eliminating the User Confounding Bias for Causal Multi-touch Attribution. *arXiv preprint arXiv:2201.00689* (2021).

[57] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. 2020. A survey on causal inference. *arXiv preprint arXiv:2002.02770* (2020).