Learning Post-Hoc Causal Explanations for Recommendation

Shuyuan Xu, Yunqi Li, Shuchang Liu, Zuohui Fu, Yongfeng Zhang
Department of Computer Science, Rutgers University, New Brunswick, NJ 08901
{shuyuan.xu,yunqi.li,shuchang.liu,zuohui.fu,yongfeng.zhang}@rutgers.edu

ABSTRACT

State-of-the-art recommender systems have the ability to generate high-quality recommendations, but usually cannot provide intuitive explanations to humans due to the usage of black-box prediction models. The lack of transparency has highlighted the critical importance of improving the explainability of recommender systems. In this paper, we propose to extract causal rules from the user interaction history as post-hoc explanations for the black-box sequential recommendation mechanisms, whilst maintain the predictive accuracy of the recommendation model. Our approach firstly achieves counterfactual examples with the aid of a perturbation model, and then extracts personalized causal relationships for the recommendation model through a causal rule mining algorithm. Experiments are conducted on several state-of-the-art sequential recommendation models and real-world datasets to verify the performance of our model on generating causal explanations. Meanwhile, We evaluate the discovered causal explanations in terms of quality and fidelity, which show that compared with conventional association rules, causal rules can provide personalized and more effective explanations for the behavior of black-box recommendation models.

KEYWORDS

Sequential Recommendation; Explainable Recommendation; Post-hoc Explanation; Causal Analysis

1 INTRODUCTION

As widely used in decision-making, recommender systems have been recognized for its ability to provide high-quality services that reduces the gap between products and customers. Nowadays, many state-of-the-art performances are achieved by neural network models. Typically, deep learning models are used as black-box latent factor models accompanied with a high-dimensional latent space. This allows them to achieve good expressiveness power and accuracy in various recommendation tasks. However, it is also true that complex neural models easily go beyond the comprehension of the majority of customers, since thousands or millions of parameters are involved. Nevertheless, it is a natural demand of human-beings to understand why a model makes a specific decision, rather than blindly accepting the results without knowing the underlying reason. As a result, providing supportive information and interpretation along with the recommendation can be helpful for both the customers and the platform, since it improves the transparency, trustworthiness, effectiveness, and user satisfaction
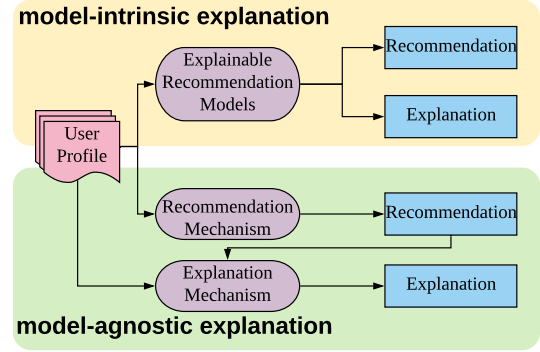
Figure 1: The model-intrinsic approaches directly generate the recommendations and explanations, while the model-agnostic approaches generate explanations after the black-box model has generated the recommendations.

of the recommendation systems, while facilitating system designers to refine the algorithms [35]. For example, a user may be inspired by the recommendation panel explained as "you may also like" in an e-commerce system, and thus decides to look around for more items that better satisfy his or her interests. In the meantime, the user preference would be captured more precisely, so that better services can be provided in further interactions. On the other hand, system designers may also easily figure out the reasons of providing unsatisfied recommendation, leveraging the result to take actions.

To address the explainability problem, researchers turned to explainable recommendation models, which are expected to not only generate effective recommendations but also intuitive explanations to humans. Generally, the explainable models can be either model-intrinsic or model-agnostic (also known as post-hoc), as shown in Fig.1. The model-intrinsic approach takes advantage of the interpretable mechanism of the model, with the explanations directly provided as the intermediate stages of recommended decisions. For instance, simple user-based collaborative filtering methods pass an item from one user to another similar user, and thus it can directly output explanations like "similar users also bought this item". However, the recommendation explanations are usually not obtained for free, and sometimes we have to trade-off with model accuracy [27]. In many cases, outstanding recommendation performances are usually achieved by models that are less interpretable. Thus, it is very challenging for current explainable recommendation methods to redesign a black-box model into an interpretable one whilst maintaining the recommendation performance [35].

In contrast, post-hoc models make no assumption of the underlying recommendation model, and allow the decision mechanism to be a black-box, since it will provide explanations after a decision is made. Although such explanations may not strictly follow the exact mechanism that generated the recommendations, they offer the flexibility to be applied to many different models. Though it is

still not fully understood what information is useful for generating explanations for a certain recommendation result, Peake [21] argued that one can provide post-hoc item-level explanations. In other words, interacted items (the causes) in a user's history can be used as explanations for the future item recommendations (the effect), since it answers "what items you bought causes the recommendation of this item".

However, existing work mostly use global association rule mining to discover the relationship between items, which relies on the item co-occurrence among all user transactions. Therefore, the explanations are not personalized, i.e., different users would receive the same explanation as long as they are recommended with the same item and have overlapped histories. This makes it incompatible with modern recommender systems, which aims to provide personalized services to users. Moreover, the item-level explanation problem naturally involves causal analysis between a user's previous and future behaviors, which makes the problem even more challenging since it has to answer counterfactual questions such as "what would happen if a different set of items were purchased".

In this paper, we explore a causal analysis framework to provide post-hoc causal explanations for any sequential recommendation algorithm. The goal is to design a model that generates post-hoc explanations for black-box recommendation models in order to reduce the accuracy-interpretability trade-off. Since the explanation model and recommendation model work separately, we obtain the benefit of explainability without hurting the prediction performance. Technically, we propose a Variational Auto-Encoder (VAE) based perturbation framework to create counterfactual examples for causal analysis, which extracts causal rules between a user's previous and future behaviors as explanations.

The key contributions of this paper are as follows:

- We design and study a causal rule mining framework for sequential recommendation.
- We show that this framework can generate personalized post-hoc explanations based on item-level causal rules to explain the behaviors of a sequential recommendation model.
- We conduct experiments on real-world data to show that our explanation model outperforms state-of-the-art baselines.

In the following, we review related work in in Section 2, and introduce our model in Section 3. Experimental settings and results are provided in Section 4. We conclude this work in Section 5.

2 RELATED WORK

2.1 Sequential Recommendation

Sequential recommendation takes into account the historical order of items interacted by a user, and aims to capture useful sequential patterns for making consecutive predictions of the user's future behaviors. Rendle et al. [24] proposed Factorized Personalized Markov Chains (FPMC) to combine Markov chain and matrix factorization for next basket recommendation. The Hierarchical Representation Model (HRM) [29] further extended this idea by leveraging representation learning as latent factors in a hierarchical model. However, these methods can only model the local sequential patterns of very limited number of adjacent records. To model multi-step sequential behaviors, He et al. [9] adopted Markov chain to provide recommendations with sparse sequences. Later on, the rapid development

of representation learning and neural networks introduced many new techniques that further push the research of sequential recommendation to a new level. For example, Hidasi et. al. [10] used an RNN-based model to learn the user history representation, Yu et. al. [34] provided a dynamic recurrent model, Li et. al. [16] proposed an attention-based GRU model, Chen et. al. [6] developed user- and item-level memory networks, and Huang et. al. [11] further integrated knowledge graphs into memory networks. However, most of the models exhibit complicated neural network architectures, and it is usually difficult to interpret their prediction results. As a result, we would like to generate explanations for these black box sequential recommendation models.

2.2 Explainable Recommendation

Explainable recommendation focuses on developing models that can generate not only high-quality recommendations but also intuitive explanations, which help to improve the transparency of the recommendation systems [35]. Generally, the explainable models can either be model-intrinsic or model-agnostic as introduced in introduction. As for model-intrinsic approaches, lots of popular explainable recommendation methods, such as factorization models [5, 28, 36], deep learning models [6, 7, 15, 25], and knowledge graph models [1, 8, 11, 19, 31, 33] have been proposed. A more complete review of the related models can be seen in [35]. However, they mix the recommendation mechanism with interpretable components, which often results in a system too complex to make successful explanations. Moreover, the increased model complexity may reduce the interpretability. A natural way to avoid this dilemma is to rely on model-agnostic post-hoc approaches so that the recommendation system is free from the noises of the down-stream explanation generator. Examples include [20] that proposed a bandit approach, [30] that proposed a reinforcement learning framework to generate sentence explanations, and [21] that developed an association rule mining approach. In their work, the transactions of all users, which consider user history as input and recommendation item as output, are used to extract association rules as the explanation for black-box models. However, correlation is not reliable for its direction agnostic feature. As will mention in the next part, our goal here is to find causal relationships in the user behaviour, which we can provide more stable explanations.

2.3 Causal Inference in Recommendation

Originated as statistical problems, causal inference [12, 22] aims at understanding and explaining the causal effect of one variable on another. While the observational data is considered as the factual world, causal effect inferences should be aware of the counterfactual world, thus often regarded as the question of "what-if". The challenge is that it is often expensive or even impossible to obtain counterfactual data. For example, it is immoral to re-do the experiment on a patient to find out what will happen if we have not given the medicine. Though the majority of causal inference study resides in the direction of statistics and philosophy, it recently attract attention from AI community for its great power of explainablity and bias elimination ability. Efforts have managed to bring causal inference to several machine learning areas, including recommendation [3], learning to rank [13], natural language processing [32], and

reinforcement learning [4], etc. With respect to recommendation tasks, large amount of work is about how to achieve de-bias matrix factorization with causal inference. The probabilistic approach ExpoMF proposed in [18] directly incorporated user exposure to items into collaborative filtering, where the exposure is modeled as a latent variable. Liang et. al.[17] followed to develop a causal inference approach to recommender systems which believed that the exposure and click data came from different models, thus using the click data alone to infer the user preferences would be biased by the exposure data. They used causal inference to correct for this bias for improving generalization of recommendation systems to new data. Bonner et. al.[3] proposed a new domain adaptation algorithm which was learned from logged data including outcomes from a biased recommendation policy, and predicted recommendation results according to random exposure. Differently, this paper focuses on learning causal rules to provide more intuitive explanation for the black-box recommendation models. Additionally, we consider [2] as a highly related work though it is originally proposed for natural language processing tasks. As we will discuss in the later sections, we utilize some of the key ideas of its model construction, and show why it works in sequential recommendation scenarios.

3 PROPOSED APPROACH

In this section, we first define the explanation problem and then introduce our model as a combination of two parts: a VAE-based perturbation model that generates the counterfactual samples for causal analysis, and a causal rule mining model that can extract causal dependencies between the cause-effect items.

3.1 Problem Setting

We denote the set of users as $\mathcal{U} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\}$ and set of items as $\mathcal{I} = \{i_1, i_2, \cdots, i_{|\mathcal{I}|}\}$. Each user u is associated with a purchase history represented as a sequence of items $\mathcal{H}^u$. The j-th interacted item in the history is denoted as $H_j^u \in \mathcal{I}$. A black-box sequential recommendation model $\mathcal{F} : \mathcal{H} \rightarrow \mathcal{I}$ is a function that takes a sequence of items (as will discuss later, it can be the permuted user history) as input and outputs the recommended item. In practice, the underlying mechanism usually consists of two steps: a ranking function that scores all candidate items based on the user history, and then selects the item with best score as the final output. Note that it only uses user-item interaction without any content or context information, and the scores predicted by the ranking function may differ according to the tasks (e.g. $\{1, \ldots, 5\}$ for rating prediction, while $[0, 1]$ for CTR prediction). Our goal is to find an item-level post-hoc model that captures the causal relation between the history items and the recommended item for each user.

DEFINITION 1. *(Causal Relation) For two variables X and Y, if X triggers Y, then we say that there is a causal relation $X \Rightarrow Y$, where X is the **cause** and Y is the **effect**.*

When a given recommendation model $\mathcal{F}$ maps a user history $\mathcal{H}^u$ to a recommended item $Y^u \in \mathcal{I}$, all items in $\mathcal{H}^u$ are considered as potential causes of Y^u. Thus we can formulate the set of causal relation candidates as $\mathcal{S}^u = \{(H, Y^u)|H \in \mathcal{H}^u\}$.

DEFINITION 2. *(Causal Explanation for Sequential Recommendation Model) Given a causal relation candidate set $\mathcal{S}^u$ for user u, if*

Algorithm 1 Causal Post-hoc Explanation Model

Input: users $\mathcal{U}$, items $\mathcal{I}$, user history $\mathcal{H}^u$, perturbation times m, black-box model $\mathcal{F}$, embedding model $\mathcal{E}$, causal mining model $\mathcal{M}$
Output: causal explanations $H \Rightarrow Y^u$ where $H \in \mathcal{H}^u$

1: Use embedding model $\mathcal{E}$ to get item embeddings $\mathcal{E}(\mathcal{I})$
2: Use $\mathcal{E}(\mathcal{I})$ and true user history to train perturbation model $\mathcal{P}$
3: **for** each user u **do**
4:     **for** i from 1 to m **do**
5:         $\tilde{\mathcal{H}}_i^u \leftarrow \mathcal{P}(\mathcal{H}^u)$; $\tilde{Y}_i^u \leftarrow \mathcal{F}(\tilde{\mathcal{H}}_i^u)$
6:     **end for**
7:     Construct perturbed input-output pairs $\{(\tilde{\mathcal{H}}_i^u, \tilde{Y}_i^u)\}_{i=1}^m$
8:     $\theta_{\tilde{H}_{ij}^u, \tilde{Y}_i^u} \leftarrow \mathcal{M}\left(\{(\tilde{\mathcal{H}}_i^u, \tilde{Y}_i^u)\}_{i=1}^m \cup (\mathcal{H}^u, Y^u)\right)$
9:     Rank $\theta_{\tilde{H}_{ij}^u, Y^u}$ and select top-k pairs $\{(H_j, Y^u)\}_{j=1}^k$
10:     **if** $\exists H_{\min\{j\}} \in \mathcal{H}^u$ **then**
11:         Generate causal explanation $H_{\min\{j\}} \Rightarrow Y^u$
12:     **else**
13:         No explanation for the recommended item Y^u
14:     **end if**
15: **end for**
16: **return** all causal explanations $H \Rightarrow Y^u$

---

there exists a true causal relation $(H, Y^u) \in \mathcal{S}^u$, then the causal explanation for recommending Y^u is described as "Because you purchased H, the model recommends you Y^u", denoted as $H \Rightarrow Y^u$.

Then the remaining problem is how to determine whether a candidate pair is a true causal relation. We can mitigate the problem by allowing a likelihood estimation for a candidate pair to be causal relation. In other words, we would like to find a ranking function that predicts the likelihood for each candidate pair. In this way, causal explanations can be generated by selecting the most promising ones from these candidates. One advantage of this formulation is that it allows the possibility that there is no causal relation between a user's history and the recommended item, e.g. when algorithm recommends the most popular items regardless of the user history. In the following sections, we will illustrate in detail how our model solves these problems.

3.2 Causal Model for Post-Hoc Explanation

We thus introduce our causal explanation framework for recommendation. Inspired by [2], we divide our framework into two models: a perturbation model and a causal rule mining model. The overview of the model framework is shown in Fig.2. Before introducing our framework in detail, we define an important concept:

DEFINITION 3. *(Causal Dependency): For a given pair of causal relation candidate (H, Y^u), the causal dependency of the pair is the likelihood of the pair being a true causal relation.*

*3.2.1 **Perturbation Model**.* To capture the causal dependency between items in history and the recommended items, we want to know what would happen if the user history had been different. To avoid unknown influences caused by the length of input sequence (i.e., user history), we keep the input length unchanged, and only replace items in the sequence to create different sequences. Ideally, for each item H_j^u in a user's history $\mathcal{H}^u$, it will be replaced by all
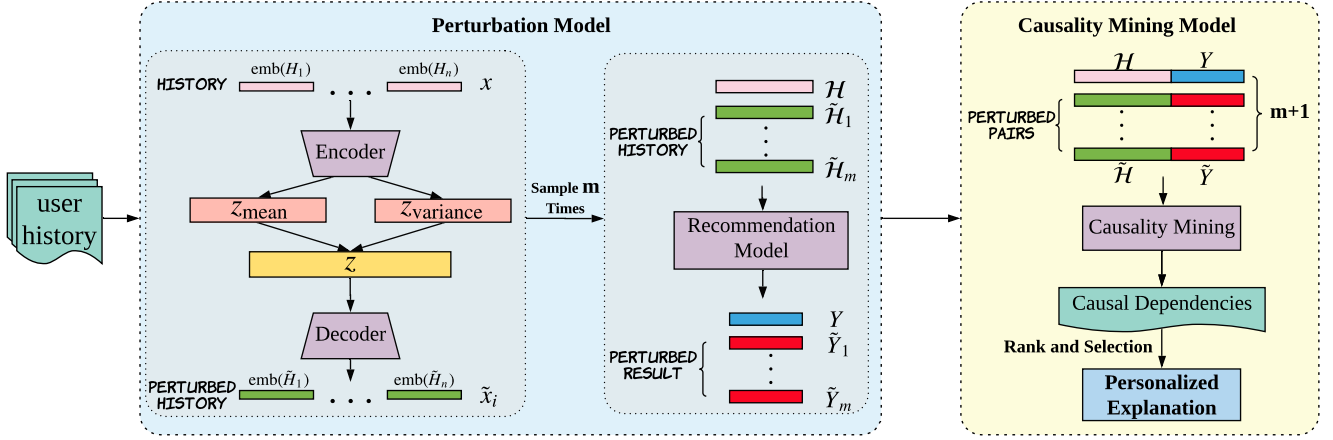
Figure 2: Model framework. x is the concatenation of the item embeddings of the user history. $\tilde{x}$ is the perturbed embedding.

possible items in $\mathcal{I}$ to fully explore the influence that H_j^u makes in the history. However, the number of possible combinations will become impractical for the learning system, since recommender systems usually deal with hundreds of thousands or even millions of items. Therefore, we pursue a perturbation-based method that samples the sequences, which randomly replaces items in the original user history $\mathcal{H}^u$.

There are various ways to obtain the perturbed version of user history, as long as they are similar to the original history. The simplest solution is randomly selecting an item in $\mathcal{H}^u$ and replacing it with a randomly selected item from $\mathcal{I}$. However, user histories are far from random selections. Instead, a natural assumption is that the user behavior follows a certain distribution. Here, we adopt VAE to learn such a distribution. As shown in Figure 2, we design a VAE-based perturbation method, which creates item sequences that are similar to but slightly different from a user's true behavior sequence, by sampling from a distribution in latent embedding space centered around the user's true sequence.

In detail, the VAE component consists of a probabilistic encoder $(\mu, \sigma) = \text{ENC}(\mathcal{X})$ and a decoder $\tilde{\mathcal{X}} = \text{DEC}(z)$. The encoder encodes a sequence of item embeddings $\mathcal{X}$ into latent embedding space, and extracts the variational information for the sequence, i.e., mean and variance of the latent embeddings under independent Gaussian distribution. The decoder generates a sequence of item embeddings $\tilde{\mathcal{X}}$ given a latent embedding z sampled from the Gaussian distribution. Here, both $\mathcal{X}$ and $\tilde{\mathcal{X}}$ are ordered concatenations of pre-trained item embeddings based on pair-wise matrix factorization (BPRMF) [23]. We follow the standard training regime of VAE by maximizing the variational lower bound of the data likelihood [14]. Specifically, the reconstruction error involved in this lower bound is calculated by a softmax across all items for each position of the input sequence. We observe that VAE can reconstruct the original data set accurately, while offering the power of perturbation.

After pretraining $\text{ENC}(\cdot)$ and $\text{DEC}(\cdot)$, the variational nature of this model allow us to obtain perturbation $\tilde{\mathcal{H}}$ for any history $\mathcal{H}$. More specifically, it first extracts the mean and variance of the encoded item sequences in the latent space, and the perturbation model samples m latent embeddings z based on the above variational information. These sampled embeddings z are then passed

to the decoder $\text{DEC}(\cdot)$ to obtain the perturbed versions $\tilde{\mathcal{X}}$. For now, each item embedding in $\tilde{\mathcal{X}}$ may not represent an actual item since it is a sampled vector from the latent space, as a result, we find its nearest neighbor in the candidate item set $\mathcal{I}$ through dot product similarity as the actual item. In this way, $\tilde{\mathcal{X}}$ is transformed into the final perturbed history $\tilde{\mathcal{H}}$. One should keep in mind that the variance should be kept small during sampling, so that the resulting sequences are similar to the original sequence.

Finally, the generated perturbed data $\tilde{\mathcal{H}}$ together with the original $\mathcal{H}$ will be injected into the black-box recommendation model $\mathcal{F}$ to obtain the recommendation results $\tilde{Y}$ and Y, correspondingly. After completing this process, we will have m perturbed input-output pairs: $\{(\tilde{\mathcal{H}}_i^u, \tilde{Y}_i^u)\}_{i=1}^m$, as well as the original pair $(\mathcal{H}^u, Y^u)$.

3.2.2 ***Causal Rule Learning Model.*** Denote $\mathcal{D}^u$ as the combined records of perturbed input-output pairs $\{(\tilde{\mathcal{H}}_i^u, \tilde{Y}_i^u)\}_{i=1}^m$ and the original pair $(\mathcal{H}^u, Y^u)$ for user u. We aim to develop a causal model that first extracts causal dependencies between input and output items appeared in $\mathcal{D}^u$, and then selects the causal rule based on these inferred causal dependencies.

Let $\tilde{\mathcal{H}}_i^u = [\tilde{H}_{i1}^u, \tilde{H}_{i2}^u, \cdots, \tilde{H}_{in}^u]$ be the input sequence of the i-th record of $\mathcal{D}^u$. Let $\tilde{Y}_i^u$ represent the corresponding output. Note that this includes the original pair $(\mathcal{H}^u, Y^u)$. The model should be able to infer the causal dependency $\theta_{\tilde{H}_{ij}^u, \tilde{Y}_i^u}$ between input item $\tilde{H}_{ij}^u$ and output item $\tilde{Y}_i^u$. We consider that the occurrence of a single output can be modeled as a logistic regression model on causal dependencies from all the input items in the sequence:

$$P(\tilde{Y}_i^u | \tilde{\mathcal{H}}_i^u) = \sigma\left(\sum_{j=1}^{n} \theta_{\tilde{H}_{ij}^u, \tilde{Y}_i^u} \cdot \gamma^{n-j} \right) \qquad (1)$$

where σ is the sigmoid function defined as $\sigma(x) = (1 + \exp(-x))^{-1}$ in order to scale the score to $[0, 1]$. Additionally, in recommendation task, the order of a user's previously interacted items may affect their causal dependency with the user's next interaction. A closer behavior tends to have a stronger effect on user's future behaviors, and behaviors are discounted if they happened earlier [10]. Therefore, we involve a weight decay parameter γ to represent the time effect. Here γ is a positive value less than one.

For an input-output pair in $\mathcal{D}^u$, the probability of its occurrence generated by Eq.(1) should be close to one. As a result, we learn the causal dependencies θ by maximizing the probability over $\mathcal{D}^u$. When optimizing θ, they are always initialized as zero to allow for no causation between two items. By learning this regression model, we are able to gradually increase θ until they converge to the point where the data likelihood of $\mathcal{D}$ is maximized.

After gathering all the causal dependencies, we select the items that have high θ scores to build causal explanations. This involves a three-step procedure.

1. We select those causal dependencies $\theta_{\tilde{H}_{ij}^u, \tilde{Y}_i^u}$ whose output is the original output Y^u (i.e., $\tilde{Y}_i^u = Y^u$). Note that these $(\tilde{H}_{ij}^u, Y^u)$ pairs may come from either the original sequence or perturbed sequences, because when a perturbed sequence is fed into the black-box recommendation model, the output may happen to be the same as the original sequence Y^u.

2. We sort the above selected causal dependencies in descending order and take the top-k $(\tilde{H}_{ij}^u, Y^u)$ pairs.

3. If there exist one or more pairs in these top-k pairs, whose cause item $\tilde{H}_{ij}^u$ appears in the user's original input sequence $\mathcal{H}^u$, then we pick such pair of the highest rank, and construct $\tilde{H}_{ij}^u \Rightarrow Y^u$ as the causal explanation for the given user. Otherwise, i.e., no cause item appears in the user history, then we output no causal explanation for the user.

Note that the extracted causal explanation is personalized since the algorithm is applied on $\mathcal{D}^u$, which only contains records centered around the user's original record $(\mathcal{H}^u, Y^u)$, while collaborative learning among users is indirectly modeled by the VAE-based perturbation model. The overall algorithm is provided in Alg.1.

4 EXPERIMENTS

In this part, we conduct experiments to show what causal relationships our model can capture and how they can serve as an intuitive explanation for the black-box recommendation model.

4.1 Dataset Description

We evaluate our proposed causal explanation framework against baselines on two datasets. The first dataset is MovieLens100k[1]. This dataset consists of information about users, movies and ratings. In this dataset, each user has rated at least 20 movies, and each movie can belong to several genres. The second dataset is the office product dataset from Amazon[2], which contains the user-item interactions from May 1996 to July 2014. The original dataset is 5-core. To achieve sequential recommendation with input length of 5, we select the users with at least 15 purchases and the items with at least 10 interactions.

Since our framework is used to explain sequential recommendation models, we split the dataset chronologically. To learn the pre-trained item embeddings based on BPRMF [23] (section 3.2.1), we take the last 6 interactions from each user to construct the testing set, and use all previous interactions from each user as the training set. To avoid data leakage, when testing the black-box recommendation models and our VAE-based perturbation model, we

Dataset	# users	# items	# interactions	# train	# test	sparsity
Movielens	943	1682	100,000	95,285	14,715	6.3%
Amazon	573	478	13,062	9,624	3,438	4.7%

Table 1: Summary of the Datasets

only use the last 6 interactions from each user (i.e., the testing set of the pre-training stage). Following common practice, we adopt the leave-one-out protocol, i.e., among the 6 interactions, we use the last one for testing, and the previous 5 as input to recommendation models. A brief summary of the data is shown in Table 1.

4.2 Experimental Settings

We adopt the following methods to train black-box sequential recommendation models and to extract traditional association rules as comparative explanations. We include both shallow and deep models for experiment.

FPMC [24]: The Factorized Personalized Markov Chain model, which combines matrix factorization and Markov chains to capture user's personalized sequential behavior patterns for prediction[3].

GRU4Rec [10]: A session-based recommendation model, which uses recurrent neural networks – in particular, Gated Recurrent Units (GRU) – to capture sequential patterns for prediction[4].

Caser [26]: The ConvolutionAl Sequence Embedding Recommendation (Caser) model, which adopts convolutional filters over recent items to learn the sequential patterns for prediction[5].

Association Rule [21]: A post-hoc explanation model, which learns the item-item association rules as item-level explanations[6].

For black-box recommendation models FPMC, GRU4Rec and Caser, we adopt their best parameter selection in their corresponding public implementation. For the association rule-based explanation model, we follow the recommendations in [21] to set the optimal parameters: support = 0.1, confidence = 0.1, lift = 0.1, length = 2 for *MovieLens100k*, and support = 0.01, confidence = 0.01, lift = 0.01, length = 2 for *Amazon* dataset due to its smaller scale.

For our causal rule learning framework, we set the item embedding size as 16, both the VAE encoder and decoder are Multi-Layer Perceptrons (MLP) with two hidden layers, and each layer consists of 1024 neurons. The default number of perturbed input-output pairs is $m = 500$ on both datasets. The default time decay factor is $\gamma = 0.7$. We will discuss the influence of perturbation times m and time decay factor γ in the experiments.

In the following, we will apply both association rule learning and causal rule learning frameworks on the black-box recommendation models to evaluate and compare the association explanations and causal explanations. In particular, we evaluate our framework from two perspectives. First, we verify that the causal rules learned by our framework represent highly probable causal relationships (explanation quality). Second, we show that our model has the ability to offer explanations for most recommendations (explanation fidelity). Additionally, we shed light on how our model differs from other models on statistical metrics.

Dataset	Movielens 100k						Amazon					
Method	Causal			Association			Causal			Association		
Parameter	$k=1$	$k=2$	$k=3$	Support	Confidence	Lift	$k=1$	$k=2$	$k=3$	Support	Confidence	Lift
FPMC	96.50%	99.25%	99.57%	14.32%	13.47%	13.47%	95.11%	98.95%	99.82%	12.57%	12.04%	12.04%
GRU4Rec	98.51%	99.36%	99.68%	7.423%	7.635%	7.635%	95.94%	99.48%	99.65%	8.202%	8.551%	8.551%
Caser	97.03%	99.15%	99.57%	8.801%	8.165%	8.165%	95.99%	99.30%	99.65%	9.250%	8.901%	8.901%

Table 2: Results of Model Fidelity. Our causal explanation framework is tested under the number of candidate causal explanations $k = 1, 2, 3$. The association explanation framework is tested under support, confidence, and lift thresholds, respectively.

4.3 Causality Verification

We first verify the quality of the extracted causal rules. Here, we adopt the following widely used definition of causation, which is introduced by Pearl [22]:

$$\Pr(\text{effect}|do(\text{cause})) > \Pr(\text{effect}|do(\neg\text{cause})) \qquad (2)$$

where $do(c)$ represents an external intervention, which compels the truth of c, and $do(\neg c)$ compels the truth of not c. Actually, the conditional probability $\Pr(e|c)$ represents a probability resulting from a passive observation of c, which rarely coincides with $\Pr(e|do(c))$. We evaluate how many percentage of our extracted causal explanations really satisfy Eq.(2).

Suppose the perturbation model (section 3.2.1) creates m perturbed input-output pairs for each user u: $\{(\tilde{\mathcal{H}}_i^u, \tilde{Y}_i^u)\}_{i=1}^m$, plus the original pair $(\mathcal{H}^u, Y^u)$. Here $\tilde{\mathcal{H}}^u$ is created by our perturbation model (i.e. not observed in the original data), and thus observing $\tilde{\mathcal{H}}^u$ implies we have $do(\tilde{\mathcal{H}}^u)$ in advance. Let $H \Rightarrow Y^u$ be the causal explanation extracted by the casual rule learning model (section 3.2.2). Then we estimate the probability based on these $m + 1$ total pairs as,

$$
\begin{aligned}
\Pr(\text{effect}|do(\text{cause})) = \Pr(Y^u|H) &= \frac{\#\text{Pairs}(H \in \tilde{\mathcal{H}}^u \wedge Y = Y^u)}{\#\text{Pairs}(H \in \tilde{\mathcal{H}}^u)} \\
\Pr(\text{effect}|do(\neg\text{cause})) = \Pr(Y^u|\neg H) &= \frac{\#\text{Pairs}(H \notin \tilde{\mathcal{H}}^u \wedge Y = Y^u)}{\#\text{Pairs}(H \notin \tilde{\mathcal{H}}^u)}
\end{aligned} \qquad (3)
$$

and $H \Rightarrow Y^u$ is considered as a reliable causal rule if Eq.(2) is satisfied. The intuition here is that a causal rule should guarantee that the probability of seeing Y^u when H is purchased should be higher than the probability of seeing Y^u when H is not purchased.

We calculate the percentage of our extracted causal explanations that satisfy Eq.(2). We tune the parameter k from 1 to 3, where k is the number of candidate causal explanations we consider for each user in the causal rule learning model (section 3.2.2 step 2 and 3). The results of causality verification on two datasets and three recommendation models are shown in Table 3.

Based on the results we can see that in most cases 90% or more of the extracted causal explanations are reliable. We also see that with the increasing of k, the percentage tends to decease for all three recommendation models on both datasets. Intuitively, this means that unreliable causal relation candidates may be introduced if too many candidate pairs are considered in the causal rule learning model, which is reasonable. However, if too few candidates are considered (i.e., k is too small), many users may not receive explanations at all (i.e., model fidelity will decrease). This will be further analyzed in the next subsection.

Dataset	Movielens			Amazon		
k	1	2	3	1	2	3
FPMC	94.83%	94.23%	93.92%	91.56%	90.48%	90.03%
GRU4Rec	97.84%	97.76%	97.75%	94.91%	94.20%	93.87%
Caser	97.15%	96.89%	96.69%	98.91%	98.25%	98.14%

Table 3: The percentage of reliable causal explanations that satisfy the inequality in Eq.(2).
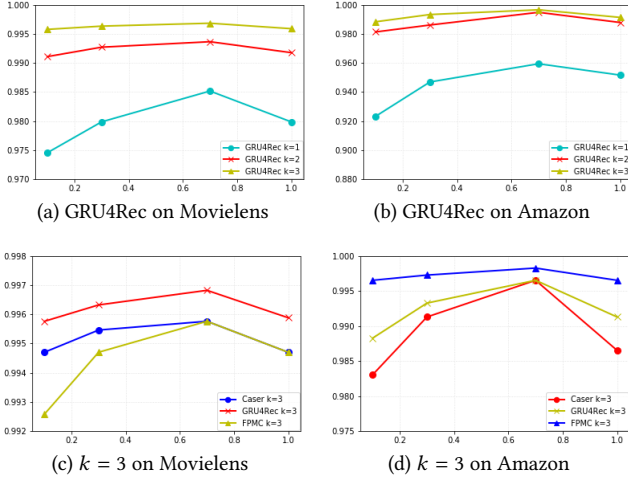
4.4 Model Fidelity

An important evaluation measure for explanation models is model fidelity, i.e., how many percentage of the recommendation results can be explained by the model [35]. The results of model fidelity are shown in Table 2. In this experiment, we still tune the number of candidate causal explanations k from 1 to 3. For the association rule explanation model (section 4.2), we test three versions of the association rule learning algorithm as introduced in [21], i.e., the association rules are filtered by support, confidence, and lift thresholds, respectively.

We can see that on both datasets, our causal explanation framework can generate explanations for almost all of the recommended items, while the association explanation approach can only provide explanations for significantly fewer recommendations. The underlying reason is that association explanations have to be extracted based on the original input-output pairs, which limits the number of pairs that we can use for rule extraction. However, based on the perturbation model, our causal explanation framework is able to create many counterfactual examples to assist causal rule learning, which makes it possible to go beyond the limited original data to extract causal explanations.

Another interesting observation is that GRU4Rec and Caser have significantly ($p < 0.01$) lower fidelity than FPMC when explained by the association model. This is reasonable because FPMC is a Markov-based model that directly learns the correlation of adjacent items in a sequence, as a result, it is easier to extract association rules between inputs and outputs for the model. However, it also means that the fidelity performance of the association approach highly depends on the recommendation model being explained. Meanwhile, we see that our causal approach achieves comparably good fidelity on all three recommendation models, because the perturbation model is able to create sufficiently many counterfactual examples to break the correlation of frequently co-occurring items in the input sequence. This indicates the robustness of our causal explanation framework in terms of model fidelity.

(a) GRU4Rec on Movielens   (b) GRU4Rec on Amazon

(c) $k = 3$ on Movielens   (d) $k = 3$ on Amazon

Figure 3: Model fidelity on different time decay parameters γ. x-axis is the time decay parameter $\gamma \in \{0.1, 0.3, 0.7, 1\}$ and y-axis is the model fidelity. The upper two sub-figures plot the model fidelity for GRU4Rec on different number of candidate causal explanations $k = 1, 2, 3$, and the lower two sub-figures plot the model fidelity for different recommendation models when fixing $k = 3$.
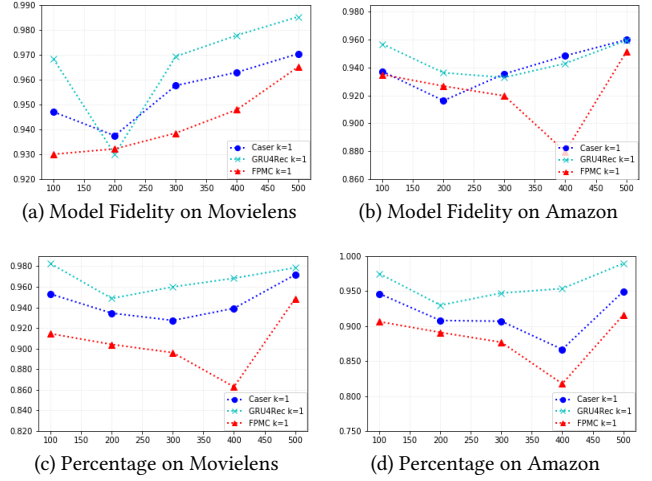
4.5 Influence of Parameters

In this section, we discuss the influence of two important parameters. The first one is time decay parameter γ – in our framework, when explaining the sequential recommendation models, earlier interactions in the sequence will have discounted effects to the recommended item. A proper time decay parameter helps the framework to reduce noise signals when learning patterns from the sequence. The second parameter is the number of perturbed input-output pairs m – in our framework, we use perturbations to create counterfactual examples for causal learning, but there may exist trade-off between efficiency and performance. We will analyze the influence of these two parameters.

Time Decay Effect: Figure 3 shows the influence of γ on different recommendation models and datasets. From the result we can see that the time decay effect γ indeed affects the model performance on fidelity. In particular, when γ is small, the previous interactions in a sequence are more likely to be ignored, which thus reduces the performance on model fidelity. When γ is large (e.g., $\gamma = 1$), old interactions will have equal importance with latest interactions, which also hurts the performance. We can see from the results that the best performance is achieved at about $\gamma = 0.7$ on both datasets.

Number of Perturbations: Figure 4 shows the influence for the number of perturbed input-output pairs m. A basic observation from Figure 4 is that when m increases, both model fidelity and the percentage of verified rules will decrease first and then increase. The underlying reason is as follows.

When m is small, the variance of the perturbed input-output pairs will be small, and thus almost any change in the input sequence will be determined as a cause. For example, suppose the original input-output pair is $A, B, C \rightarrow Y$. In the extreme case where $m = 1$,


(a) Model Fidelity on Movielens   (b) Model Fidelity on Amazon

(c) Percentage on Movielens   (d) Percentage on Amazon

Figure 4: Model fidelity and the percentage of verified causal explanations (Eq.(2)) on different number of perturbed pairs m. x-axis is the number of perturbed pairs m. y-axis is model fidelity. The upper two sub-figures plot the model fidelity for different recommendation model on $k = 1$, and the lower two sub-figures plot the percentage of verified causal explanations for different recommendation model on $k = 1$.

we will have only one perturbed pair, e.g., $A, \tilde{B}, C \rightarrow \tilde{Y}$. According to the causal rule learning model (section 3.2.2), if $\tilde{Y} \neq Y$, then $B \Rightarrow Y$ will be the causal explanation since the change of B results in a different output, while if $\tilde{Y} = Y$, then either $A \Rightarrow Y$ or $C \Rightarrow Y$ will be the causal explanation since their θ scores will be higher than B or $\tilde{B}$. In either case, the model fidelity and percentage of verified causal rules will be 100%. However, in this case, the results do not present statistical meanings since they are estimated on a very small amount of examples.

When m increases but not large enough, then random noise examples created by the perturbation model will reduce the model fidelity. Still consider the above example, if many pairs with the same output Y are created, then the model may find other items beyond A, B, C as the cause, which will result in no explanation for the original sequence. However, if we continue to increase m to sufficiently large numbers, such noise will be statistically offset, and thus the model fidelity and percentages will increase again. In the most ideal case, we would create all of the $|\mathcal{H}|^{|\mathcal{I}|}$ sequences for causal rule learning, where $|\mathcal{H}|$ is the number of item slots in the input sequence, and $|\mathcal{I}|$ is the total number of items in the dataset. However, $|\mathcal{H}|^{|\mathcal{I}|}$ would be a huge number that makes it computational infeasible for causal rule learning. In practice, we only need to specify m sufficiently large. Based on Chebyshev's Inequality, we find that $m = 500$ already gives >95% confidence that the estimated probability error is <0.1.

4.6 Case Study

In this section, we provide a simple qualitative case study to compare causal explanations and association explanations. Compared with the association explanation model, our model is capable of generating personalized explanations, which means that even if
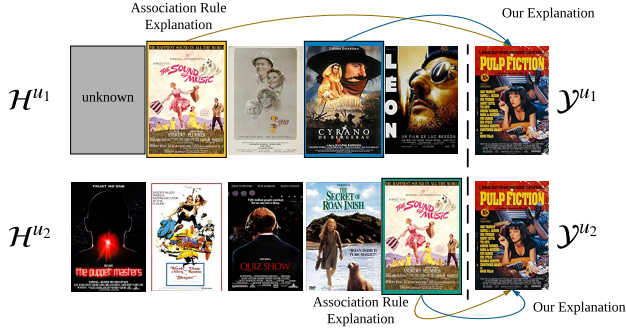
Figure 5: A case study on MovieLens by the Caser model. The first movie for u_1 is unknown in the dataset.

the recommendation model recommends the same item for two different users and the users have overlapped histories, our model still has the potential to generate different explanations for different users. However, the association model will provide the same explanation since the association rules are extracted based on global records. An example by the Caser recommendation model on *Movie-Lens100k* dataset is shown in Figure 5, where two users with one commonly watched movie (*The Sound of Music*) get exactly same recommendation (*Pulp Fiction*). The association model provides the overlapped movie as an explanation for the two different users, while our model can generate personalized explanation for different users even when they got the same recommendation.

5 CONCLUSIONS

Recommender systems are widely used in our daily life. Effective recommendation mechanisms usually work through black-box models, resulting in the lack of transparency. In this paper, we extract causal rules from user history to provide personalized, item-level, post-hoc explanations for the black-box sequential recommendation models. The causal explanations are extracted through a perturbation model and a causal rule learning model. We conduct experiments on real-world datasets, and apply our explanation framework to several state-of-the-art sequential recommendation models. Experimental results verified the quality and fidelity of the causal explanations extracted by our framework.

In this work, we only considered item-level causal relationships, while in the future, it would be interesting to explore causal relations on feature-level external data such as textual user reviews, which can help to generate finer-grained causal explanations.

ACKNOWLEDGEMENT

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* 11, 9 (2018), 137.

[2] David Alvarez-Melis and Tommi S Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017).

[3] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM.

[4] Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. 2019. Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search. In *International Conference on Learning Representations*.

[5] Jingwu Chen, Fuzhen Zhuang, Xin Hong, Xiang Ao, Xing Xie, and Qing He. 2018. Attention-driven factor model for explainable personalized recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 909–912.

[6] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 108–116.

[7] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. 2018. Automatic generation of natural language explanations. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*. ACM, 57.

[8] Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, et al. 2020. Fairness-Aware Explainable Recommendation over Knowledge Graphs. *SIGIR* (2020).

[9] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 191–200.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*.

[11] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 505–514.

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

[13] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 781–789.

[14] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

[15] Chenliang Li, Cong Quan, Li Peng, Yunwei Qi, Yuming Deng, and Libing Wu. 2019. A Capsule Network for Recommendation and Explaining What You Like and Dislike. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 275–284.

[16] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1419–1428.

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

[18] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*. 951–961.

[19] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *The World Wide Web Conference*. 1210–1221.

[20] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 31–39.

[21] Georgina Peake and Jun Wang. 2018. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[22] Judea Pearl. 2000. *Causality: models, reasoning and inference*. Vol. 29. Springer.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *UAI* (2012).

[24] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 811–820.

[25] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 297–305.

[26] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 565–573.

[27] Andreas Theodorou, Robert H Wortham, and Joanna J Bryson. 2017. Designing and implementing transparency for real time inspection of autonomous robots. *Connection Science* 29, 3 (2017), 230–241.

[28] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 165–174.

[29] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 403–412.

[30] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018. A Reinforcement Learning Framework for Explainable Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 587–596.

[31] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5329–5336.

[32] Zach Wood-Doughty, Ilya Shpitser, and Mark Dredze. 2018. Challenges of Using Text Classifiers for Causal Inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4586–4598.

[33] Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 285–294.

[34] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 729–732.

[35] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* (2020).

[36] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 83–92.