Learning Causal Explanations for Recommendation

Shuyuan Xu[1], Yunqi Li[1], Shuchang Liu[1], Zuohui Fu[1], Yingqiang Ge[1], Xu Chen[2] and Yongfeng Zhang[1]

[1]Department of Computer Science, Rutgers University, New Brunswick, NJ 08901, US
[2]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, 100872, China

Abstract

State-of-the-art recommender systems have the ability to generate high-quality recommendations, but usually cannot provide 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 construct causal explainable recommendation which aims to provide post-hoc explanations for the recommendations by answering "what if" questions, e.g., "what would the recommendation result change if the user's behavior history had been different?" Our approach first obtains counterfactual user histories and counterfactual recommendation items with the aid of a perturbation model, and then extracts personalized causal relationships for the recommendation model through a causal rule mining algorithm. Different from some existing explainable recommendation models that aim to provide persuasive explanations, our model aims to find out the true explanations for the recommendation of an item. Therefore, in addition to evaluating the fidelity of discovered causal explanations, we adopt the average causal effect to measure the quality of explanations. Here by quality we mean whether they are true explanations rather than their persuasiveness. We conduct experiments for several state-of-the-art sequential recommendation models on real-world datasets to verify the performance of our model on generating causal explanations.

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 reduce the gap between products and customers. And many state-of-the-art models achieves outstanding expressiveness by using high-dimensional user/item representations and deep learning models with thousands or even millions of parameters [1, 2]. However, this excessive complexity easily go beyond the comprehension of a human who may demand for intuitive explanations for why the model made a specific decision. Moreover, providing supportive information and interpretation along with the recommendation can be helpful for both the customers and the platform, since it improves the transparency, persuasiveness, trustworthiness, effectiveness, and user satisfaction of the recommendation systems, while facilitating system designers to refine the algorithms [3]. Thus, people are looking for solutions that can generate explanations along with the recommendation.

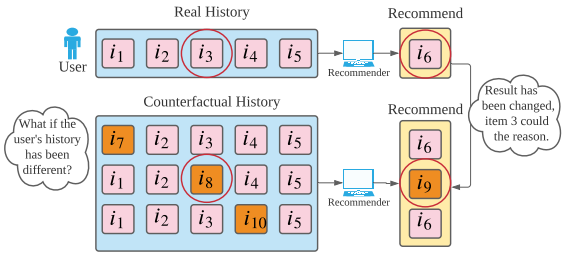


Figure 1: An example of causal explanation. Comparing the recommendation of real history and counterfactual histories, if replacing one certain item will result in the change of recommendation, the certain item could be the true reason that the system recommends the original item.

One typical method to solve explainable recommendation is to construct a model-intrinsic explanation module that also serves as an intermediate recommendation stage[4, 5]. However, this approach has to redesign the original recommendation model and thus may sacrifice model accuracy in order to obtain good explanations [6]. Moreover, for complex deep models, it is even more challenging to integrate an explainable method into the original design while maintaining recommendation performance [3]. In contrast, post-hoc models (a.k.a model-agnostic explanation) consider the underlying recommendation model as a black-box, and provide explanations

The 1st International Workshop on Causality in Search and Recommendation (CSR'21), July 15, 2021, Virtual Event, Canada
shuyuan.xu@rutgers.edu (S. Xu); yunqi.li@rutgers.edu (Y. Li); shuchang.syt.liu@rutgers.edu (S. Liu); zuohui.fu@rutgers.edu (Z. Fu); yingqiang.ge@rutgers.edu (Y. Ge); xu.chen@ruc.edu.cn (X. Chen); yongfeng.zhang@rutgers.edu (Y. Zhang)
https://zuohuif.github.io/ (Z. Fu); https://yingqiangge.github.io/ (Y. Ge); http://xu-chen.com/ (X. Chen); http://yongfeng.me (Y. Zhang)

after the recommendation decision has been made. Although such explanations may not strictly follow the exact mechanism that generated the corresponding recommendations, they offer the flexibility to be applied to a wide range of recommendation models. Furthermore, the explanation model and recommendation model work separately, we obtain the benefit of explainability without hurting the prediction performance.

While it is still not fully understood what information is useful to generate the explanations for a certain recommendation result, Peake [7] argued that one can provide post-hoc item-level explanations. Specifically, interacted items (the causes) in a user's history can be used as explanations for the future item recommendations (the effect).The authors propose to solve this by association rule mining which finds co-occurred items as explanation. However, explanations generated by association rules are not personalized, i.e., different users would receive the same explanation as long as the rules are only applied to their overlapped histories. This makes it incompatible with modern recommender systems, which aim to provide personalized services to users. Moreover, we believe that the true explanation of a recommendation model should be able to answer the questions like "which item contribute to the system's decision?" as well as " Will the system change the decision if a different set of items were purchased by the same user? " In other words, the explanation should be aware of the counterfactual world of the unobserved user histories and their corresponding recommendation when analyzing the cause of a recommendation in real world.

In this paper, we explore a counterfactual analysis framework to provide post-hoc causal explanations for any given black-box sequential recommendation algorithm. Fig.1 shows an example to illustrate our intuition. Technically, we first create several counterfactual histories which are different but similar to the real history through a Variational Auto-Encoder (VAE) based perturbation model, and obtain the recommendation for the counterfactual data. Then we apply causal analysis on the combined data to extract causal rules between a user's history and future behaviors as explanations. Unlike other explainable recommendation models [4, 8, 9] that focus on persuading users to keep engaged with the system, this type of explanation focuses on model transparency and finds out the true reason or the most essential item that leads to a specific recommendation. Therefore, instead of taking user studies or online evaluations to evaluate the persuasiveness or effectiveness of explanations, we use the average causal effect to measure whether the item used for explanation can explain how the system works.

The key contributions of this paper are as follows:

- We design and study a counterfactual explainable framework for a wide range of sequential recommendations.
- We show that this framework can generate personalized post-hoc explanations based on item-level causal rules.
- We conduct several experiments on real-world data to demonstrate that our explanation model outperforms state-of-the-art baselines in terms of fidelity.
- We apply average causal effect to illustrate that the causal explanations provided by our framework are essential component for most sequential recommendation model.

For the remainder of this paper, we first review related work in Section 2, and then introduce our model in Section 3. Experimental settings and results are provided in Section 4. Finally, 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 to make consecutive predictions of the user's future behaviors. Rendle et al. [10] proposed Factorized Personalized Markov Chains (FPMC) to combine Markov chain and matrix factorization for next basket recommendation. The Hierarchical Representation Model (HRM) [11] 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. [12] 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 pushed the research of sequential recommendation to a new level. For example, Hidasi et. al. [13] used an RNN-based model to learn the user history representation, Yu et. al. [14] provided a dynamic recurrent model, Li et. al. [15] proposed an attention-based GRU model, Chen et. al. [16] developed user- and item-level memory networks, and Huang et. al. [17] 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. To make up for this, we plan 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 [3]. Generally, the explainable models can be either model-intrinsic or model-agnostic. As for model-intrinsic approaches, lots of popular explainable recommendation methods, such as factorization models [4, 18, 9, 19], deep learning models [20, 16, 21, 22], knowledge graph models [23, 5, 17, 24, 25, 26], explanation ranking models [27], logical reasoning models [1, 28, 29], dynamic explanation models [30, 31], visual explanation models [8] and natural language generation models [32, 33, 34] have been proposed. A more complete review of the related models can be seen in [3]. However, they mix the recommendation mechanism with interpretable components, which often results in over-complicated systems 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 [35] that proposed a bandit approach, [36] that proposed a reinforcement learning framework to generate sentence explanations, and [7] that developed an association rule mining approach. Additionally, some work distinguish the model explanations by their purpose [37]: while persuasive explanations aim to improve user engagement, model explanation reflexes how the system really works and may not necessarily be persuasive. Our study fall into the later case and aims to find causal explanations for a given sequential recommendation model.

2.3. Causal Inference in Recommendation

Originated as statistical problems, causal inference [38, 39] 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 being regarded as the questions 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 has recently attracted the 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 [40], learning to rank [41], natural language processing [42], and reinforcement learning [43], 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 [44] directly incorporated user exposure to items into collaborative filtering, where the exposure is modeled as a latent variable. Liang et. al. [45] 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. [40] 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. Besides de-bias recommendation, Ghazimatin et. al. [46] proposed PRINCE model to explore counterfactual evidence for discovering causal explanations in a heterogeneous information network. Differently, this paper focuses on learning causal rules to provide more intuitive explanation for the black-box sequential recommendation models. Additionally, we consider [47] 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 $H_j^u \in \mathcal{I}$. Without specification, the calligraphic $\mathcal{H}$ in the paper represents user history, and a straight H represents an item. 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 counterfactual user history) as input and outputs the recommended item. In practice, the underlying mechanism usually consists of two steps: a ranking function first scores all candidate items based on the user history,

and then it selects the item with the highest 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 Click Through Rate (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 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 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 being a causal relation.

Definition 3. *(Causal Dependency) For a given candidate pair of causal relation (H, Y^u), the causal dependency θ_{H,Y^u} of that pair is the likelihood of the pair being a true causal relation.*

In other words, we would like to find a ranking function that predicts the likelihood for each candidate pair, and the causal explanation is generated by selecting the pair with top ranking score from these candidates. One advantage of this formulation is that it allows the possibility of giving 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.

3.2. Causal Model for Post-Hoc Explanation

In this section, we introduce our counterfactual explanation framework for recommendation. Inspired by [47], 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.

3.2.1. Perturbation Model

To capture the causal dependency between items in history and the recommended items, we want to know what would take place 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 counterfactual histories. Ideally, for each item H_j^u in a user's history $\mathcal{H}^u$, it will be replaced by all 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 tens of millions items. In fact, counterfactual examples that are closest to the original input can be the most useful to a user as shown in [48]. Therefore, we pursue a perturbation-based method that generate counterfactual examples, which replaces items in the original user history $\mathcal{H}^u$.

There are various ways to obtain the counterfactual history, as long as they are similar to the real history. The simplest solution is randomly selecting an item in $\mathcal{H}^u$ and replacing it with a randomly selected item from $\mathcal{I} \setminus \mathcal{H}^u$. However, user histories are far from random. Thus, we assume that their exists a ground truth user history distribution, and we adopt VAE to learn such a distribution. As is 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 genuine history sequence, by sampling from a distribution in the latent embedding space centered around the user's true history sequence.

In detail, the VAE component consists of a probabilistic encoder $(\mu, \sigma) = \mathrm{ENC}(\mathcal{X})$ and a decoder $\tilde{\mathcal{X}} = \mathrm{DEC}(z)$. The encoder $\mathrm{ENC}(\cdot)$ takes 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 $\mathrm{DEC}(\cdot)$ 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 (BPR-MF) [49]. We follow the standard training regime of VAE by maximizing the variational lower bound of the data likelihood [50]. 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 $\mathrm{ENC}(\cdot)$ and $\mathrm{DEC}(\cdot)$, the variational nature of this model allows us to obtain counterfactual history $\tilde{\mathcal{H}}$ for any real history $\mathcal{H}$. More specifically, it first extracts the mean and variance of the encoded item

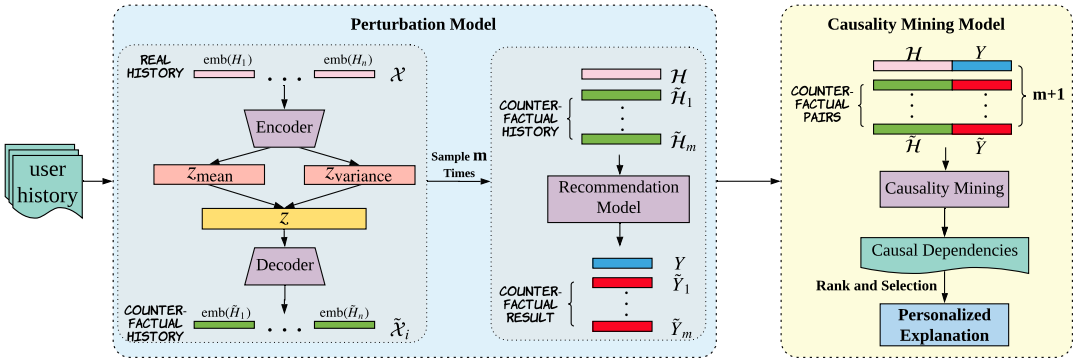


Figure 2: Model framework. x is the concatenation of the item embeddings of the user history. $\tilde{x}$ is the perturbed embedding.

sequences in the latent space, and then 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 $\mathcal{I} \setminus \mathcal{H}$ through dot product similarity as the actual item. In this way, $\tilde{\mathcal{X}}$ is transformed into the final counterfactual history $\tilde{\mathcal{H}}$. One should keep in mind that the variance should be kept small during sampling, so that the resulting sequences can be similar to the original sequence.

Finally, the generated counterfactual 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. For any user u, after completing this process, we will have m different counterfactual input-output pairs: $\{(\tilde{\mathcal{H}}_i^u, \tilde{Y}_i^u)\}_{i=1}^m$, as well as the original pair $(\mathcal{H}^u, Y^u)$. Here the value of m is manually set, but it cannot exceed the number of all possible item combinations.

3.2.2. Causal Rule Learning Model

Denote $\mathcal{D}^u$ as the combined records of counterfactual 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 outputs items appeared in $\mathcal{D}^u$, and then selects the causal rule based on these inferred causal dependencies.

Let $\hat{\mathcal{H}}_i^u = [\hat{H}_{i1}^u, \hat{H}_{i2}^u, \cdots, \hat{H}_{in}^u]$ be the input sequence of the i-th record of $\mathcal{D}^u$, where $\hat{H}_{ij}^u$ is the j-th item in $\hat{\mathcal{H}}_i^u$. Let $\hat{Y}_i^u$ represent the corresponding output. Note that this includes the original real pair $(\mathcal{H}^u, Y^u)$. The model should be able to infer the causal dependency (refer to Definition 3) $\theta_{\hat{H}_{ij}^u, \hat{Y}_i^u}$ between input item $\hat{H}_{ij}^u$ and output item $\hat{Y}_i^u$. We consider that the occurrence of a single output can be modeled as a logistic regression on causal dependencies from all the input items in the sequence:

$$P(\hat{Y}_i^u | \hat{\mathcal{H}}_i^u) = \sigma\Big(\sum_{j=1}^{n} \theta_{\hat{H}_{ij}^u, \hat{Y}_i^u} \cdot \gamma^{n-j}\Big) \tag{1}$$

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 [13]. 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. When learning this regression model, we are able to gradually increase θ until they converge to the point where the data likelihood of $\mathcal{D}^u$ 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 $\theta_{\hat{H}_{ij}^u, \hat{Y}_i^u}$ whose output is the original Y^u (i.e., $\hat{Y}_i^u = Y^u$). Note that these $(\hat{H}_{ij}^u, Y^u)$ pairs may come from either the original sequence or counterfactual sequences, because when a counterfactual sequence is fed into the black-box recommendation model, the output may happen to be the same as the original sequence Y^u.

Algorithm 1 Causal Explanation Model

Input: users $\mathcal{U}$, items $\mathcal{I}$, user history $\mathcal{H}^u$, counterfactual number 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 counterfactual input-output pairs $\{(\tilde{\mathcal{H}}_i^u, \tilde{Y}_i^u)\}_{i=1}^m$
8: $\{(\hat{\mathcal{H}}_i^u, \hat{Y}_i^u)\}_{i=1}^{m+1} \leftarrow \{(\tilde{\mathcal{H}}_i^u, \tilde{Y}_i^u)\}_{i=1}^m \cup (\mathcal{H}^u, Y^u)$
9: $\theta_{\hat{H}_{ij}^u, \hat{Y}_i^u} \leftarrow \mathcal{M}\left(\{(\hat{\mathcal{H}}_i^u, \hat{Y}_i^u)\}_{i=1}^{m+1}\right)$
10: Rank $\theta_{\hat{H}_{ij}^u, Y^u}$ and select top-k pairs $\{(H_j, Y^u)\}_{j=1}^k$
11: **if** $\exists H_{\min\{j\}} \in \mathcal{H}^u$ **then**
12: Generate causal explanation $H_{\min\{j\}} \Rightarrow Y^u$
13: **else**
14: No explanation for the recommended item Y^u
15: **end if**
16: **end for**
17: **return** all causal explanations $H \Rightarrow Y^u$

2. We sort the above selected causal dependencies in descending order and take the top-k $(\hat{H}_{ij}^u, Y^u)$ pairs.
3. If there exist one or more pairs in these top-k pairs, which cause item $\hat{H}_{ij}^u$ appears in the user's input sequence $\mathcal{H}^u$, then we pick such pair of the highest rank, and construct $\hat{H}_{ij}^u \Rightarrow Y^u$ as the causal explanation for the 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. For each user, there are two phases - perturbation phase (line 4-7) and causal rule mining phase (line 8-15).

4. Experiments

In this section, 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.

Table 1
Summary of the Datasets

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%

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. Further, to learn the pre-trained item embeddings based on BPR-MF [49] (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 only use the last 6 interactions of 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 in test set, we use the last one for testing, and the previous five interactions will serve as input to the 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 extract traditional association rules as comparative explanations. Meanwhile, we further conduct different variants of the perturbation model to analyze our model. We include both shallow and deep models for the experiment.

FPMC [10]: The Factorized Personalized Markov Chain model, which combines matrix factorization and Markov chains to capture user's personalized sequential behavior patterns for prediction[3].

[1] https://grouplens.org/datasets/movielens/
[2] https://nijianmo.github.io/amazon/
[3] https://github.com/khesui/FPMC

Table 2
Results of Model Fidelity. Our causal explanation framework is tested under the number of candidate causal explanations $k = 1$. The association explanation framework is tested under support, confidence, and lift thresholds, respectively. The best fidelity on each column is highlighted in bold.

Dataset	Movielens 100k				Amazon			
Models	FPMC	GRU4Rec	NARM	Caser	FPMC	GRU4Rec	NARM	Caser
AR-sup	0.3160	0.1453	0.4581	0.1569	0.2932	0.1449	0.4066	0.2024
AR-conf	0.2959	0.1410	0.4305	0.1559	0.2949	0.1449	0.4031	0.1885
AR-lift	0.2959	0.1410	0.4305	0.1559	0.2949	0.1449	0.4031	0.1885
CR-AE	0.5631	0.7413	0.7084	0.6151	0.6981	0.8255	0.8970	0.7260
CR-VAE	**0.9650**	**0.9852**	**0.9714**	**0.9703**	**0.9511**	**0.9721**	**0.9791**	**0.9599**

GRU4Rec [13]: A session-based recommendation model, which uses recurrent neural networks – in particular, Gated Recurrent Units (GRU) – to capture sequential patterns for prediction[4].

NARM [15]: A sequential recommendation model which utilizes GRU and attention mechanism to estimate the importance of each interactions[5].

Caser [51]: The ConvolutionAl Sequence Embedding Recommendation (Caser) model, which adopts convolutional filters over recent items to learn the sequential patterns for prediction[6].

AR-sup [7]: A post-hoc explanation model, which extract association rules from interactions from all users and rank based on support value to generate item-level explanations.

AR-conf [7]: Extracting association rules and rank based on confidence value to get explanations.

AR-lift [7]: Rank based on lift value among extracted association rules to generate explanations.

CR-AE: A variant of our causal rule model which applies fixed variance in hidden layer of AutoEncoder model as the perturbation model. Compared with our VAE-based perturbation model, this variant apply non-personalized variance.

For black-box recommendation models FPMC, GRU4Rec, NARM 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 [7] 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. We accept top 100 rules based on corresponding values (i.e. support/confidence/lift) as explanations

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 only difference between our model and the variant model CR-AE is that the variant model applies fixed normal distribution as variance instead of learned personalized variance. The default number of counterfactual input-output pairs is $m = 500$ on both datasets. The default time decay factor is $\gamma = 0.7$. We will discuss the influence of counterfactual number m and time decay factor γ in the experiments.

In the following, we will apply our model and all baselines on the black-box recommendation models to evaluate and compare the generated explanations. In particular, we evaluate our framework from three perspectives. First, a explanation model should at least be able to offer explanations for most recommendations, we will show it in the result (explanation fidelity). Second, if our model is capable of generating explanations for most recommendations, we need to verify that the causal explanations learned by our framework represent the key component of recommendation mechanism (explanation quality). Finally, since counterfactual examples are involved in our framework, our framework should be able to generate closer counterfactual examples (counterfactual quality). Additionally, we shed light on how our model differs from other models on statistical metrics.

4.3. Model Fidelity

A very basic purpose of designing a explanation model is to generate explanations for most recommendations. Therefore, an important evaluation measure for explanation models is model fidelity, i.e., what's the percentage of the recommendation results can be explained by the model [3]. The results of model fidelity are shown in Table 2. In this experiment, we only report the results of keeping the number of candidate causal explanations k as 1 for our framework and variant. For the association rule explanation model (section 4.2), we apply the global association rules [7] ranking by support, confidence, and lift, respectively.

[4] https://github.com/hungthanhpham94/GRU4REC-pytorch
[5] https://github.com/Wang-Shuo/Neural-Attentive-Session-Based-Recommendation-PyTorch
[6] https://github.com/graytowne/caser_pytorch

We can see that on both datasets, our causal explanation framework is able to generate explanations for most of the recommended items (including the variant), 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 capable of creating many counterfactual examples to assist causal rule learning, which makes it possible to go beyond the limited original data to extract causal explanations. Moreover, when the number of input and output items are limited (e.g. five history items as input and the model recommends only one item in our case), it is harder to match global rules with personal interactions and recommendation, which limits the flexibility of global association rules.

Another interesting observation is that GRU4Rec and Caser have significantly ($p < 0.01$) lower fidelity than FPMC and NARM when explained by the association model. This is reasonable because FPMC is a Markov-based model that consider input as a basket and directly learns the correlation between candidate items and each items in a sequence, as a result, it is easier to extract association rules between inputs and outputs for the model. NARM combines the whole session information and influence of each individual item in the session, therefore, association rules which involve individual information will be easier to be extracted for this 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 sufficient 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.

4.4. Average Causal Effect

We then verify our causal explanations are true explanations that explanation are important component for recommending original item. A common way is to measure the causal effect on the outcome of the model[52]. First of all, we show the definition of Average Causal Effect.

Definition 4. *(Average Causal Effect) The **Average Causal Effect** (ACE) of a binary random variable x on another random variable y is define as* $\mathbb{E}[y|do(x=1)] - \mathbb{E}[y|do(x=0)]$

Here $do()$ represents an external intervention, which forces a variable to take a specific value. Specifically, in our case, for an extracted causal rule $H \Rightarrow Y^u$, we define the binary random variable as 1 if $H \in \tilde{\mathcal{H}}_i^u$, 0 else. We also define another variable y as a binary random variable, which is 1 if $\tilde{Y}_i^u = Y_u$, otherwise it will be 0. We then report average ACE on all generated explanations. Note that since the ACE value is used for causal related models, we cannot report it on the association rule baseline.

Suppose the perturbation model (section 3.2.1) creates m counterfactual input-output pairs for each user u: $\{(\tilde{\mathcal{H}}_i^u, \tilde{Y}_i^u)\}_{i=1}^m$. Here $\tilde{\mathcal{H}}^u$ is created by our perturbation model (i.e. not observed in the original data), and thus observing $H \in \tilde{\mathcal{H}}^u$ implies we have $do(x=1)$ 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 ACE based on these m counterfactual pairs as,

$$\begin{aligned}
\mathbb{E}[y|do(x=1)] &= \Pr(y=1|do(x=1)) \\
&= \frac{\#\text{Pairs}(H \in \tilde{\mathcal{H}}^u \wedge Y = Y^u)}{\#\text{Pairs}(H \in \tilde{\mathcal{H}}^u)} \\
\mathbb{E}[y|do(x=0)] &= \Pr(y=1|do(x=0)) \\
&= \frac{\#\text{Pairs}(H \notin \tilde{\mathcal{H}}^u \wedge Y = Y^u)}{\#\text{Pairs}(H \notin \tilde{\mathcal{H}}^u)}
\end{aligned} \tag{2}$$

We report the ACE value of our model and variants in Table.3. While showing the ACE value, we still keep the number of candidate causal explanations k as 1.

We can see that our model can achieve higher ACE value than the variant for most recommendation models on both dataset. But here we can observe an interesting results that the ACE value for FPMC model is much lower than other recommendation models (GRU4Rec, NARM, Caser). Meanwhile, the variant model has slightly larger ACE than our model when applying on FPC model.

The difference between FPMC and other recommendation models is that FPMC is based on Markov chain that only considers the last behavior while other models involve the whole session information. For FPMC model, although we take a session as input to recommend next item, this model actually considers it as a basket and linearly combines the influence of each item from the basket. In this case, every part of the session will have independent influence towards next item prediction. So changing a small part of input items may not significantly change the next item prediction which high likely results in same recommendation item. Based on our experiment, when we keep counterfactual histories same for all recommendation models, FPMC model only gets 98 counterfactual histories (19.6%) in average with different recommendation (different from the recommendation item based on real history), while other models have at least 315 counterfactual histories (63%) in average with different recommendation item. This difference

Table 3
Results of Average Causal Effect. Our causal explanation framework is tested under the number of candidate causal explanations $k = 1$.

Dataset	Movielens 100k			
Models	FPMC	GRU4Rec	NARM	Caser
CR-AE	0.0184	0.1479	0.1108	0.1199
CR-VAE	0.0178	0.1862	0.1274	0.1388
Dataset	Amazon			
Models	FPMC	GRU4Rec	NARM	Caser
CR-AE	0.0230	0.1150	0.1101	0.1347
CR-VAE	0.0212	0.1434	0.1511	0.1563

makes the FPMC model has much lower ACE value compared with other recommendation models. Comparing our model with CR-AE, the variant model will generate less similar counterfactual histories which more likely result in different recommendation item than our model. Therefore, CR-AE has slightly higher ACE values than CR-VAE.

4.5. Proximity

As we mentioned before, counterfactual examples that are closest to the original can be the most useful to users. Similar with [48], we define the proximity as the distance between negative counterfactual examples (i.e. generate recommendation item different from original item) and original real history. Intuitively, a counterfactual example that close enough but get totally different results will be more helpful. For a given user, the proximity can be expressed as

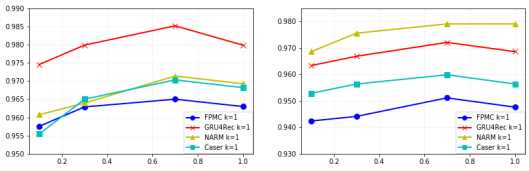
$$Proximity_u = -mean(\sum_{\tilde{Y}_i^u \neq Y_u} dist(\tilde{\mathcal{H}}_i^u, \mathcal{H}^u)) \quad (3)$$

Here the distance is defined in latent space. The representation of any history sequence is the concatenation of the latent representation of each item in the sequence. The latent representations of items are learned from pretrained BPRMF [49] model. The distance of any two sequence is defined as Euclidean distance between the representation of two sequence. The reported proximity value would be the average over all users.

Given that association rule model does not involve counterfactual examples, this metric can only be reported on our model and the variant model on both datasets, as shown in Table.4 We can observe that our model can achieve higher proximity compared with the variant model. In other words, counterfactual examples generated with learned latent variance is more similar with real history. Therefore, higher proximity implies counterfactual examples of our model have higher quality and be more useful.

Table 4
Results of Proximity. The value of proximity is calculated by Eq.(3)

Dataset	Movielens 100k			
Models	FPMC	GRU4Rec	NARM	Caser
CR-AE	-22.69	-22.37	-22.35	-22.40
CR-VAE	-17.35	-16.88	-16.83	-16.93
Dataset	Amazon			
Models	FPMC	GRU4Rec	NARM	Caser
CR-AE	-21.83	-21.28	-21.20	-21.33
CR-VAE	-18.01	-17.40	-17.31	-17.51



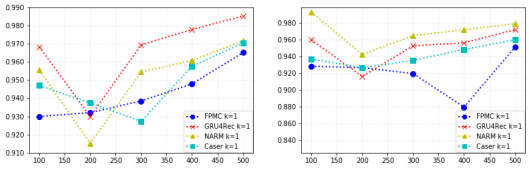
(a) Model Fidelity on Movielens (b) Model Fidelity on Amazon

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 left side pictures are on *Movielens* and the right side pictures are on *Amazon*.

4.6. 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



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

Figure 4: Model fidelity on different number of counterfactual pairs m. x-axis is the number of counterfactual pairs m. y-axis is model fidelity.

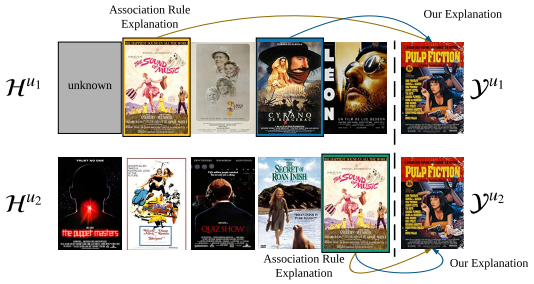


Figure 5: A case study on MovieLens by the Caser model. The first movie for u_1 is unknown in the dataset.

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 Counterfactual Examples: Figure 4 shows the influence for the number of counterfactual input-output pairs m. A basic observation from Figure 4 is that when m increases, model fidelity will decrease first and then increase. The underlying reason is as follows.

When m is small, the variance of the counterfactual input-output pairs will be small, and fewer counterfactual items will be involved. Then the model is more likely to select original item as explanation. For example, suppose the original input-output pair is $A, B, C \rightarrow Y$. In the extreme case where $m = 1$, we will have only one counterfactual 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.

4.7. Case Study

In this section, we provide a simple 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 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 [51] recommendation model on *MovieLens100k* 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 several 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.

Acknowledgments

This work was partly supported by NSF IIS-1910154 and IIS-2007907. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

[1] H. Chen, S. Shi, Y. Li, Y. Zhang, Neural collaborative reasoning, in: Proceedings of the Web Conference 2021, 2021, pp. 1516–1527.

[2] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys (CSUR) 52 (2019) 1–38.

[3] Y. Zhang, X. Chen, Explainable recommendation: A survey and new perspectives, Foundations and Trends® in Information Retrieval (2020).

[4] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma, 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, 2014, pp. 83–92.

[5] Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo, Y. Zhang, Reinforcement knowledge graph reasoning for explainable recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 285–294.

[6] A. Theodorou, R. H. Wortham, J. J. Bryson, Designing and implementing transparency for real time inspection of autonomous robots, Connection Science 29 (2017) 230–241.

[7] G. Peake, J. Wang, 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, 2018.

[8] X. Chen, H. Chen, H. Xu, Y. Zhang, Y. Cao, Z. Qin, H. Zha, Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 765–774.

[9] N. Wang, H. Wang, Y. Jia, Y. Yin, Explainable recommendation via multi-task learning in opinionated text data, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018.

[10] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, in: Proceedings of the 19th international conference on World wide web, ACM, 2010, pp. 811–820.

[11] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, X. Cheng, Learning hierarchical representation model for nextbasket recommendation, in: Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2015, pp. 403–412.

[12] R. He, J. McAuley, Fusing similarity models with markov chains for sparse sequential recommendation, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 191–200.

[13] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, in: International Conference on Learning Representations, 2016.

[14] F. Yu, Q. Liu, S. Wu, L. Wang, T. Tan, 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, 2016, pp. 729–732.

[15] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, J. Ma, Neural attentive session-based recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 1419–1428.

[16] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, H. Zha, Sequential recommendation with user memory networks, in: Proceedings of the eleventh ACM international conference on WSDM, 2018, pp. 108–116.

[17] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen, E. Y. Chang, Improving sequential recommendation with knowledge-enhanced memory networks, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018, pp. 505–514.

[18] J. Chen, F. Zhuang, X. Hong, X. Ao, X. Xie, Q. He, Attention-driven factor model for explainable personalized recommendation, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018, pp. 909–912.

[19] X. Chen, Z. Qin, Y. Zhang, T. Xu, Learning to rank features for recommendation over multiple categories, in: Proceedings of the 39th International ACM SIGIR conference on Research and Develop-

ment in Information Retrieval, 2016, pp. 305–314.
[20] S. Seo, J. Huang, H. Yang, Y. Liu, Interpretable convolutional neural networks with dual local and global attention for review rating prediction, in: Proceedings of the Eleventh ACM Conference on RecSys, 2017, pp. 297–305.
[21] C. Li, C. Quan, L. Peng, Y. Qi, Y. Deng, L. Wu, 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, 2019, pp. 275–284.
[22] F. Costa, S. Ouyang, P. Dolog, A. Lawlor, Automatic generation of natural language explanations, in: Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion, ACM, 2018, p. 57.
[23] Q. Ai, V. Azizi, X. Chen, Y. Zhang, Learning heterogeneous knowledge base embeddings for explainable recommendation, Algorithms 11 (2018) 137.
[24] Z. Fu, Y. Xian, R. Gao, J. Zhao, Q. Huang, Y. Ge, S. Xu, S. Geng, C. Shah, Y. Zhang, et al., Fairness-aware explainable recommendation over knowledge graphs, SIGIR (2020).
[25] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, X. Ren, Jointly learning explainable rules for recommendation with knowledge graph, in: The World Wide Web Conference, 2019, pp. 1210–1221.
[26] Y. Xian, Z. Fu, H. Zhao, Y. Ge, X. Chen, Q. Huang, S. Geng, Z. Qin, G. De Melo, S. Muthukrishnan, et al., Cafe: Coarse-to-fine neural symbolic reasoning for explainable recommendation, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1645–1654.
[27] L. Li, Y. Zhang, L. Chen, Extra: Explanation ranking datasets for explainable recommendation, SIGIR (2021).
[28] S. Shi, H. Chen, W. Ma, J. Mao, M. Zhang, Y. Zhang, Neural logic reasoning, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1365–1374.
[29] Y. Zhu, Y. Xian, Z. Fu, G. de Melo, Y. Zhang, Faithfully explainable recommendation via neural logic reasoning, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 3083–3090.
[30] X. Chen, Y. Zhang, Z. Qin, Dynamic explainable recommendation based on neural attentive models, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 53–60.
[31] Q. Ai, Y. Zhang, K. Bi, W. B. Croft, Explainable product search with a dynamic relation embedding model, ACM Transactions on Information Systems (TOIS) 38 (2019) 1–29.
[32] L. Li, Y. Zhang, L. Chen, Personalized transformer for explainable recommendation, ACL (2021).
[33] L. Li, Y. Zhang, L. Chen, Generate neural template explanations for recommendation, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 755–764.
[34] H. Chen, X. Chen, S. Shi, Y. Zhang, Generate natural language explanations for recommendation, SIGIR 2019 Workshop on ExplainAble Recommendation and Search (2019).
[35] J. McInerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson, R. Mehrotra, Explore, exploit, and explain: personalizing explainable recommendations with bandits, in: Proceedings of the 12th ACM Conference on Recommender Systems, ACM, 2018, pp. 31–39.
[36] X. Wang, Y. Chen, J. Yang, L. Wu, Z. Wu, X. Xie, A reinforcement learning framework for explainable recommendation, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 587–596.
[37] N. Tintarev, Explanations of recommendations, in: Proceedings of the 2007 ACM conference on Recommender systems, 2007, pp. 203–206.
[38] J. Pearl, Causality: models, reasoning and inference, volume 29, Springer, 2000.
[39] G. W. Imbens, D. B. Rubin, Causal inference in statistics, social, and biomedical sciences, Cambridge University Press, 2015.
[40] S. Bonner, F. Vasile, Causal embeddings for recommendation, in: Proceedings of the 12th ACM Conference on Recommender Systems, ACM, 2018.
[41] T. Joachims, A. Swaminathan, T. Schnabel, Unbiased learning-to-rank with biased feedback, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 781–789.
[42] Z. Wood-Doughty, I. Shpitser, M. Dredze, Challenges of using text classifiers for causal inference, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4586–4598.
[43] L. Buesing, T. Weber, Y. Zwols, S. Racaniere, A. Guez, J.-B. Lespiau, N. Heess, Woulda, coulda, shoulda: Counterfactually-guided policy search, in: ICLR, 2019.
[44] D. Liang, L. Charlin, J. McInerney, D. M. Blei, Modeling user exposure in recommendation, in: Proceedings of the 25th WWW, 2016.
[45] D. Liang, L. Charlin, D. M. Blei, Causal inference for recommendation, in: Causation: Foundation to Application, Workshop at UAI, 2016.
[46] A. Ghazimatin, O. Balalau, R. Saha Roy, G. Weikum,

Prince: provider-side interpretability with counterfactual explanations in recommender systems, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 196–204.

[47] D. Alvarez-Melis, T. S. Jaakkola, 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).

[48] R. K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020.

[49] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, UAI (2012).

[50] D. P. Kingma, M. Welling, Auto-encoding variational bayes, 2014.

[51] J. Tang, K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, ACM, 2018, pp. 565–573.

[52] R. Moraffah, M. Karami, R. Guo, A. Raglin, H. Liu, Causal interpretability for machine learning-problems, methods and evaluation, ACM SIGKDD Explorations Newsletter 22 (2020) 18–33.