Counterfactual Explainable Recommendation Systems

Abstract

Recommender systems are crucial for helping users discover relevant items and enhancing user engagement. A key challenge is providing clear explanations for recommendations. In this project, we implement (Tan et al., 2021) counterfactual approach to explainable recommendations, which uses causal inference to generate simple, interpretable explanations.

The paper formulates a joint optimization problem that minimizes the change in item features to reverse a recommendation. These altered features explain the recommendation. The paper also introduces two evaluation metrics for explainable recommender systems, considering both user and model perspectives.

1 Introduction

Explainability has become increasingly important in recommendation systems, prompting the development of Counterfactual Explainable Recommendation (CER) projects. Advanced recommendation techniques like LFM and deep learning improve accuracy but often lack interpretability. Explainable systems aim to provide personalized recommendations with intuitive explanations, enhancing user trust, satisfaction, and overall experience. Various approaches for explainability in recommender systems include:

1. Model-based methods: These methods employ models capable of generating explanations for recommendations. However, they can be limited by their complexity and lack of scalability.

2. Post-hoc methods: These methods generate explanations after recommendations have been made, but can be constrained by their interpretability and lack of personalization.

The paper implemented in this project focuses on aspect-aware explanations using counterfactual reasoning, emphasizing the complexity and strength of the explanations to generate simple yet effective explanations.

The primary contributions of this project are as follows:

1. Implementing the CountER framework for a different dataset to validate its effectiveness and generalizability.

2. Investigating the base recommender EFM (Explainable Factorization Machine) model, which can serve as a foundation for further exploration of explainable recommendation approaches.

2 Related Work

In this section, we explore existing research on Explainable Recommendations and Counterfactual Reasoning.

2.1 Explainable Recommendation

Explainable recommendation research focuses on two key aspects: the explainability of recommendation methods and recommendation results. Transparent, interpretable models are developed to make the recommendation process more understandable. Approaches include factorization-based, topic modeling, graph-based, deep learning, knowledge-based, and rule mining. Some research only emphasizes the explainability of recommendation results, treating the recommendation model as a black box and developing separate models to explain its results.

One approach, the explicit Factor Model (Zhang et al., 2014a) recommends items based on user-liked features. It extracts item features from user reviews and aligns latent features with item features to generate explanations. (Chen et al., 2016) extended this idea to tensor factorizations. Wang et al. further generalized MF models to multi-task learning over tensors. (Bauman et al., 2017) proposed Sentiment Utility Logistic Model (SULM), which utilizes the features and sentiments on the features in MF. Other works extend to graph-based topic modeling, deep learning, knowledge graphs, and rule mining approaches. Most approaches identify the best features using user preference and item performance for explanation. The paper we implemented, however, uses counterfactual reasoning by

tweaking features to determine when an item would not have been recommended.

2.2 Counterfactual Reasoning

Counterfactual reasoning involves mental representations of alternative possibilities to actual events. In recommendation, recent works use counterfactual reasoning to improve accuracy and explainability based on heterogeneous information networks, perturbation models, or influence functions. The authors' method generates counterfactual explanations on the item side based on item aspects and adopts a counterfactual learning framework driven by Occam's Razor Principle to learn explanations with small complexity and large strength directly.

3 Preliminaries

We have a user set $\mathcal{U} = \{u_1, u_2, \cdots, u_n\}$, an item set $\mathcal{I} = \{i_1, i_2, \cdots, i_m\}$ and a binary matrix $\mathcal{B} \in \{0,1\}^{nXm}$ which depicts whether a user has interacted with a particular item or not.

Using the Sentires toolkit (Zhang et al., 2014b), first, we extract the feature, opinion, and score for each user and its interacted items from the textual reviews. We get features $\mathcal{F} = \{f_1, f_2, \cdots, f_f\}$. Assuming that if a user cares more about some feature, then the user will comment more frequently about those features in their reviews, and using this information the User Attention Matrix $\mathcal{X}^{nXf}$ is constructed which measures how much a user cares about each particular feature. For each user, we first check all their review [F,O,S] triplets to count how frequently the user mentions each feature. Then the User-feature Attention Matrix is computed as follows:

$$\mathcal{X}_{i,f} = \begin{cases} 0, \textit{ if user i didn't mention feature f} \\ 1 + (N-1)\left(\frac{2}{1+exp(-t_{i,f})} - 1\right) \end{cases}$$

where, N is the rating scale and $t_{i,f}$ represents how frequently user u mentioned feature f.

Similarly, we also need to compute how well each item scores on each of the features and the average sentiment on the item's feature. For each item, we check all the reviews about that item, to compute frequency and the sentiment of the features w.r.t item. Then, the Item-Feature Quality Matrix $\mathcal{Y}^{mXf}$ is computed as follows:

$$\mathcal{Y}_{j,f} = \begin{cases} 0, \textit{if item j didn't get mention on feature f} \\ 1 + (\frac{(N-1)}{1+exp(-t_{j,f} * s(j,f)} \end{cases}$$

where, N is the rating scale, $t_{j,f}$ represents how frequently item i was mentioned for feature f and $s_{j,f}$ represents the average sentiment on the feature f for item i.

We have a function that gives us $\mathcal{R}^{iXk}$, which lists top - k recommendations for user u.

3.1 Counterfactual Explainability

For an item j in a user's recommendation list $\mathcal{R}(i,k)$, we want to identify a slight change in the item's features such that the item is no longer in the top-K recommendation $\mathcal{R}(i,k)$.

1. j is in a user's recommendation list $\mathcal{R}(i,k)$
2. $\mathcal{Y}_j$ is the item's feature vector.
3. Small change in the item's feature vector $\Delta = \{\delta_1, \delta_2, \cdots, \delta_f\}$
4. Such that $\mathcal{Y}_j = \mathcal{Y}_j + \Delta$, then $j \notin \mathcal{R}(i,k)$

To optimize Δ, we need to consider the Explanation Complexity and Explanation Strength.

Explanation complexity defines how complex the explanation is, it can be divided into how many features are used to generate the explanation and how much change needs to be made on these features. The paper defines Explanation Complexity as

$$C(\Delta) = ||\Delta||_2 + \gamma||\Delta||_0$$

where $||\Delta||_0$ represents the number of non-zero values,i.e, the number of features changed and $||\Delta||_2$ represents the amount of change applied.

Further, explanation strength defines how much the change affects the ranking score. The paper defines Explanation Strength as

$$S(\Delta) = s_{i,j} - s_{i,j_\Delta}$$

where $s_{i,j}$ is the ranking score before change and s_{i,j_Δ} is the ranking score after the change. After optimizing Δ we get the explanations in the form "If the item was worse on [aspect/aspects], it wouldn't have been recommended."

4 Problem Formulation

4.1 Base Recommender model

The paper employs a deep neural network as a black-box model, using X and Y matrices to compute ranking scores. The network features a fusion layer to concatenate user and item feature vectors,

followed by three fully connected layers and a sigmoid layer, which outputs a scaled ranking score.

To further explore the base model, we experimented with different layers and feature sizes in the MLP architecture. We found that an MLP with five layers and 8192 initial features in the base layer performed better than one with three layers and 1024 initial features. Adding more layers or changing the number of features beyond that led to overfitting and increased loss. Considering training time, we concluded with the new model with 8192 initial features, as adding more parameters resulted in increased training time.

4.2 Counterfactual Model

For a given user u we get a list of top-K recommendations $\mathcal{R}(u, K)$. To obtain the explanation for an item i in the list, the optimization is based on a single principle that defines a model with minimum explanation complexity such that explanation strength is good which is mathematically described as:

$$\text{min. } C(\Delta) = ||\Delta||_2 + \gamma||\Delta||_0$$

$$\text{s.t.} \quad S(\Delta) = s_{i,j} - s_{i,j_\Delta} \geq \epsilon$$

For the threshold ϵ we will take it as the ranking score of K+1th recommended item, since we aim to change the features of item i, so that it is no longer in the top-K list, thus a score below K+1th score guarantees that item i will on longer be in the list.

$$\epsilon = s_{i,j} - s_{i,jK+1}$$

$$\text{min. } ||\Delta||_2 + \gamma||\Delta||_0 \quad \text{s.t.} \quad s_{i,j_\Delta} \leq s_{i,jK+1}$$

But the above equation suffers from non-differentiablity and non-convexity, the paper suggests the following changes :

1. Replacing 0-norm with l1-norm, which helps minimize the explanation complexity.

2. Changing the constraint to hinge loss.

$$L(s_{i,j}, s_{i,jK+1}) = max0, \alpha + s_{i,j_\Delta} - s_{i,jK+1}$$

3. Adding the hinge loss as a Lagrangian term.

We get,

$$\text{min. } ||\Delta||_2 + \gamma||\Delta||_1 + \lambda(L(s_{i,j_\Delta}, s_{i,jK+1})$$

where

$$s_{i,j_\Delta} = f(X_i, Y_j + \Delta/Z, \theta)$$

$$s_{i,j_{K+1}} = f(X_i, Y_{jK+1}/Z, \theta)$$

$$L(s_{i,j}, s_{i,jK+1}) = max\{0, \alpha + s_{i,j_\Delta} - s_{i,jK+1}\}$$

λ, α are hyper-parameters which we can tune to control the strength of Explanations.

5 Dataset Description

We utilized two new datasets: the beauty 5-core and video game 5-core from the Amazon datasets. The 5-core datasets ensure each user and item has k reviews. The beauty dataset contains 198,502 product reviews, and the video games dataset has 497,577 reviews. We first extracted sentiments from these datasets and used them for the base recommendation system. We attempted training with the Yelp Dataset using the improved recommendation model, but due to its size, training was not completed in time.

Dataset	Users	Items	Features	Reviews	Density
Beauty	2659	9423	581	48450	18.59
Video Games	6144	14980	2015	120610	19.63

6 Evaluations

The paper utilizes aspects and features from user reviews as ground truth to measure accuracy. It evaluates the results from two perspectives: user-oriented and model-oriented.

6.1 User Oriented Evaluations

User-oriented metrics evaluate recommendation results based on why the user purchased the item, using ground truth reviews. From a user's textual review, we extract aspects mentioned with positive sentiment, creating a binary vector $P_{i,j}$. The model provides a vector Δ representing non-zero changes in aspects that contribute to the explanation. $I(\delta)$ is an identity function that returns 1 if δ is non-zero, and 0 if it's zero. Using these, we measure the following metrics:

$$\text{Precision} = \frac{\sum_{k=1}^{r} p_{i,j}^{(k)} * I(\delta_k)}{\sum_{k=1}^{r} I(\delta_k)}$$

$$\text{Recall} = \frac{\sum_{k=1}^{r} p_{i,j}^{(k)} * I(\delta_k)}{\sum_{k=1}^{r} p_{i,j}^{(k)}}$$

$$F_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Precision measures the percentage of aspects in our generated explanation that the user truly liked, while Recall measures the percentage of aspects

liked by the user that are included in our explanation. The implementation also calculates the F_1 score as the harmonic mean between Precision and Recall.

6.2 Model Oriented Evaluations

Model-oriented evaluations focus on understanding why a recommendation model suggests a specific item for a user. The paper introduces metrics Probability of Necessity (PN) and Probability of Sufficiency (PS). PN score assesses the probability that an item would not be recommended if specific aspects $A_{i,j}$ didn't exist in a counterfactual world. PS score measures the probability that an item would still be recommended if only certain aspects $A_{i,j}$ existed in a counterfactual world. The implementation calculates F1, the harmonic mean between PN and PS scores.

$$\text{PN} = \frac{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in R_{i,K}} PN_{ij}}{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in R_{i,K}} I(\mathcal{A}_{ij} \neq \emptyset)}$$

$$\text{PS} = \frac{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in R_{i,K}} PS_{ij}}{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in R_{i,K}} I(\mathcal{A}_{ij} \neq \emptyset)}$$

where,

$$PN_{ij} = \begin{cases} 1, \textit{if} \quad v_j^* \notin R_{i,K}^* \\ 0, \text{else} \end{cases}$$

$$PS_{ij} = \begin{cases} 1, \textit{if} \quad v_j^* \in R_{i,K}^{`} \\ 0, \text{else} \end{cases}$$

7 Results

The implementation generates a dictionary as output, mapping the $(user_{id}, item_{id})$ pair to an array of features contributing to the explanation. The following results were obtained for both datasets using the original base recommendation model and the improved model.

7.1 User Oriented Results

The results for the user oriented evaluations were measured on the metrics described in the previous section.

Model	Dataset	Precision	Recall	F_1
Base	Beauty	0.22	0.37	0.24
Base	Video Games	0.21	0.19	0.17
Improved	Beauty	0.23	0.41	0.26
Improved	Video Games	0.22	0.20	0.17

Overall, we can see that the newly improved base recommendation model performed better in all criteria in both the datasets.

7.2 Model Oriented Results

Similary, using the previously defined metrics we report the results as below.

Model	Dataset	PN	PS	F_1
Base	Beauty	0.68	0.91	0.78
Base	Video Games	0.50	0.69	0.58
Improved	Beauty	0.68	0.94	0.79
Improved	Video Games	0.51	0.76	0.61

On the model's aspect as well the newly improved base recommendation model performed better in all criteria.

8 Conclusion

In conclusion, the implementation of the Counterfactual Explainable Recommendation (CountER) approach demonstrates a novel and effective way to provide personalized recommendations along with counterfactual explanations.

We found a direct correlation between the base recommendation system's recommendations and the explanations generated, implying that improved recommendations lead to better explanations based on product review ground truth.

Future work could explore counterfactual changes in other information types, such as images or textual descriptions, and extend CountER to explainable decision-making processes using knowledge graphs or graph neural networks.

References

Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 2017. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. *23rd ACM SIGKDD International Conference*, pages 717–725.

Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to rank features for recommendation over multiple categories. In *SIGIR*.

Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. 2021. Counterfactual Explainable Recommendation. *30th ACM International Conference*, page 1784–1793.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014a. Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis. *SIGIR*, pages 83–92.

Yongfeng Zhang, Haochen Zhang, Min Zhang, Yiqun Liu, et al. 2014b. Do Users Rate or Review? Boost Phrase-level Sentiment Labeling with Review-level Sentiment Classification. *SIGIR*, pages 1027–1030.