Post-hoc Explanations for Complex Model Recommendations using Simple Methods

Dorin Shmaryahu
Ben-Gurion University of the Negev, Israel
dorins@post.bgu.ac.il

Guy Shani
Ben-Gurion University of the Negev, Israel
shanigu@bgu.ac.il

Bracha Shapira
Ben-Gurion University of the Negev, Israel
bshapira@bgu.ac.il

ABSTRACT

Many leading approaches for generating recommendations, such as matrix factorization and autoencoders, compute a complex model composed of latent variables. As such, explaining the recommendations generated by these models is a difficult task. In this paper, instead of attempting to explain the latent variables, we provide post-hoc explanations for why a recommended item may be appropriate for the user, by using a set of simple, easily explainable recommendation algorithms. When the output of the simple explainable recommender agrees with the complex model on a recommended item, we consider the explanation of the simple model to be applicable. We suggest both simple collaborative filtering and content based approaches for generating these explanations. We conduct a user study in the movie recommendation domain, showing that users accept our explanations, and react positively to simple and short explanations, even if they do not truly explain the mechanism leading to the generated recommendations.

Author Keywords

Recommender Systems, Explainable Recommendation, content-base explanations, collaborative filtering explanations, user-study

INTRODUCTION

Recommendation systems that suggest items to users can be found in many modern applications, from online newspapers and movie streaming applications, to e-commerce [2, 26, 18]. Research has shown that in many applications, user may be interested in understanding why is a particular recommended item appropriate for her [27, 11, 31]. Thus, it is beneficial to be able to generate explanations for the recommended items.

Early simple recommendation algorithms often yield a natural explanation for their recommendations. For example, the recommendations of a neighborhood based collaborative filtering approach [20] can be explained as: “users similar to you often choose this item”. Item-item collaborative filtering algorithms [23, 3] provide recommendations that can be explained as “users who choose the item that you have chosen often also choose the recommended item”. Content-based algorithms [17], that learn for each user a set of content features that the user prefers, generate recommendations that can be explained by “the recommended item has a content feature that you prefer”.

However, these simple algorithms often provide recommendations of lower accuracy than modern approaches. In recent years, two collaborative filtering approaches became popular for generating good recommendations — the matrix factorization (MF) approach [13, 14, 15], and the artificial neural network (ANN) approach [29]. Algorithms of these families have shown the capacity to generate accurate recommendations for users.

One of the downsides of both approaches is that they compute the recommendations through a set of latent variables and their possibly non-linear relations. For example, in the MF approach one computes a vector of latent variables for each user, and a vector of latent variables for each item, and then computes a recommendation score using the inner product between the vectors of a particular user and a particular item. The values of the latent variables do not have an understandable meaning to humans.

Several researchers have attempted to provide explanations by understanding the behavior of the latent variables [32, 6]. Such efforts may be possible in some cases, but it is unlikely that all, or even most, latent variables represent an easy to understand structure. The problem becomes even more difficult with deep ANNs, that may contain thousands of such variables with complex connections between them.

Alternatively, one can take a *post-hoc* approach to explanations [12, 4], that takes the model recommendations as input, and attempts to identify reasons as to why these recommended items are appropriate to the user. For example, [21] used association rule mining to identify explanations for the recommendations directly from the data. These explanations cannot be considered to be *transparent* [28], as they do not shed light on the choices made within the model in recommending the particular item, but may still provide value to the user. They can be *effective*, helping the user in making decisions. They may be *persuasive*, convincing the user to explore the recommended item. They may also increase *trust*, by, e.g., providing a reasonable explanation for a recommendation that the user dislikes.

IntRS '20 - Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, September 26, 2020, Virtual Event

In this paper we also take a post-hoc explanation generation approach. Given the output of any black-box recommender, we run a set of easy-to-explain recommendation algorithms, such as the simple collaborative filtering and content based methods suggested above. These algorithms provide a score for the items recommended by the black box recommender. When this score is sufficiently high, it means that the explainable recommender agrees with the black box recommender. In this case, we can present the explanation of the explaining recommender to the user.

Our approach is model agnostic — we can generate explanations for any recommender. Our approach is also flexible, in that the explanations can be generated post-hoc by any easy-to-explain recommendation algorithm that outputs a recommendation score for each item. Although in this paper we study only the simple recommenders mentioned above, given any other easy-to-explain recommender, one can use it to generate new explanations, that would be candidate explanations for the items recommended by the black box recommender.

We study the user perception of explanations generated by simple easy-to-explain recommenders for the items recommended by complex models. We evaluate the user's response to recommended items with and without explanations of different types. We also measure participant user preference over the various types of explanations. To study these questions we conduct a user study in the movie domain. We use two popular recommendation models, an MF and an autoencoder, as black boxes to generate recommendations. For each recommended item we run a set of 6 easy-to-explain approaches to produce explanations for the recommendation — item-item content based, user-item content based, item-item collaborative filtering, user-user collaborative filtering, movie overview textual similarity, and a popularity recommender. We show only explanations which are sufficiently relevant, that is, whose score passes a method-dependant threshold.

We first ask participants to rank the generated recommendations without any explanation. Then, we ask their opinion about recommended items with explanation, showing a single, randomly chosen, explanation for every movie.

In the next stage of the user study, the participants were shown additional recommended movies. In this stage we presented all explanations that passed a threshold to the participants, and asked them to rate each explanation. The results in this stage show that participants preferred content based explanations to collaborative filtering explanations, and that popularity explanations are rated the lowest.

Finally, the participants completed an online survey, asking their opinion about recommendation explanations in general. Our results indicate that participants prefer short and easy to understand explanations to transparent explanations that fully disclose the mechanism behind the computed recommendations.

BACKGROUND

Recommender systems actively suggest items to users, to help them to rapidly discover relevant items, and to increase item consumption [22]. Such systems can be found in many applications, including TV streaming services [2], online e-commerce [26], smart tutoring [8], and many more [18]. We focus here one important recommendation task [24] — top-*N* recommendation, where the system computes a list of *N* recommended items that the user may choose.

There are two dominant approaches for computing recommendations for the *active user* — the user that is currently interacting with the application and the recommender system. First, the *collaborative filtering* approach [5, 10] assumes that users who agreed on preferred items in the past will tend to agree in the future too. Many such methods rely on a matrix R of user-item ratings to predict unknown matrix entries, and thus to decide which items to recommend.

A simple method in this family [20], commonly referred to as *user-user collaborative filtering*, identifies a neighborhood of users that are similar to the *active user*. A common method for computing user similarity is the Jaccard correlation $Jaccard(u_1,u_2) = \frac{I_{u_1} \cap I_{u_2}}{I_{u_1} \cup I_{u_2}}$ where I_u is the set of items consumed by a user u. This set of neighbors is based on the similarity of observed preferences between these users and the active user. Then, items that were preferred by users in the neighborhood are recommended to the active user. Another approach [23, 3], known as *item-item collaborative filtering* rely on the set of users that consumed two items i_1 and i_2. One can compute, e.g., the Jaccard correlation between the items: $Jaccard(i_1,i_2) = \frac{U_{i_1} \cap U_{i_2}}{U_{i_1} \cup U_{i_2}}$ where U_i is the set of users who consumed item i. Then, the system can recommend to a user u an item i_2 that has high Jaccard similarity to an item i_1 that u has previously consumed.

A second popular approach is known as *content-based* recommendation [17]. In this approach, the system has access to a set of item features. The system then learns the user preferences over features, and uses these computed preferences to recommend new items with similar features. Such recommendations are typically titled "similar items".

In content based recommendations one can again take an item-item approach, computing the similarity between items based on shared feature values, such as the leading actors, the same director. or the same genre. Then, one can recommend an item that has high similarity to an item that was previously consumed by the user. One can also take a user-item approach, by computing a user profile — the set of feature values that often appear in items consumed by the user, such as actors that repeatedly appear in movies that the user has consumed, or genres the the user often watches. Then, one can compute the similarity of an item to the user profile to decide whether to recommend the item to the user.

It is widely agreed in the recommendation system research community that in many domains, collaborative filtering approaches produce better recommendations than content based methods.

A collaborative filtering approach that has gained much attention in the recommender system community is the matrix factorization [13, 14, 15], where the system attempts to factor the rating matrix $R_{|U|\times|I|}$ into two matrices, $P_{|U|\times k}$ and $Q_{k\times|I|}$, for some small number k, where $R \approx P \times Q$. One can consider the matrix P as a set of latent user features, and Q as a set of latent item features. An item i is considered to be appropriate for a user u when the inner product $p_u \cdot q_i$ is high. The resulting latent feature vectors p_u and q_i typically do not have a meaning that can be translated into content features, such as actors or genres, but are associated with the user like-dislike pattern of items. As such, explaining to the user why a particular item was recommended to her, beyond the vague statement that the system predicts that the item is a good match for the user, is difficult.

Another state of the art collaborative filtering approach is the variational autoencoder (VAE). An autoencoder (AE) neural network is an unsupervised learning algorithm, attempting to produce target values equal to the input values, $y^{(i)} = x^{(i)}$. The autoencoder tries to learn a function $h_{W,b}(x) \approx x$ where W and b is the set of weights and biases corresponding to the hidden units in the deep network.

While the input and output layers of the network are large, there is an inner low dimensional layer within the network. Thus, the network learns a lower dimension representation of the input, the latent space. The autoencoder operates in two phases, an encoder that reduces the input into a compact representation in the low dimension layer, and a decoder, responsible for reconstructing the encoded representation into the original input.

In the recommendation system task, the input is a user partial item choice vector $r^{(u)}$, e.g., a vector of all movies in the system, where only movies that the user has watched receive a value of 1. The reconstruction of the input at the output layer contains higher scores for items that the user is likely to choose.

RELATED WORK

Explainable recommendations provided to users may help them understand why certain items are appropriate for them. By clarifying these reasons, explanations can improve the transparency, persuasiveness, effectiveness, trustworthiness, and user satisfaction from the recommender system [27, 11, 31]. While earlier recommenders were often naturally explainable, modern models are more complex, and do not yield natural explanations. Studies in explainable recommendations hence address the challenge of providing human understandable explanations for items recommended by complex models.

There are two main approaches to providing explainable recommendations [31]. The first approach attempts to create interpretable recommendation models whose results can be naturally explained. However, many modern models are often not naturally explainable, and making them more explainable, often results in reduced recommendation accuracy. This line of research therefore aims at mitigating the trade-off between accuracy and explainability by including explainable components, layers or external information into non-linear complex and deep accurate models to make them explainable. Examples of such solutions for MF-based recommendation models include the work by [32], who applied sentiment analysis over user reviews, to learn users preferences related features of items that served as a basis for latent feature tables.

Additional examples can be found for deep learning recommendation models, such as the work by [6], that learned the distribution of user attention over features of different items that serve as explanations. These algorithms try to analyse the meaning of each latent component in a neural network, and how they interact with each other to generate the final results.

The second approach is post-hoc and model-agnostic [12, 4]. It treats the model as a black box and explains the recommendation results in a rational way by identifying relations between the data provided as input to the recommender system and its recommended items. This analysis is decoupled from the recommendation model, considering only the model input and output. The post-hoc approach has the advantage of enabling explanations in scenarios where the recommendation model cannot be exposed. Although the post-hoc explanations presented to a user are not transparent, i.e., they do not reflect the computation used by the underlying model to provide recommendations, they commonly present rationale, plausible information for the user.

Some post-hoc explainable recommendation models use statistical methods to analyze the influence of the input on the output [7]. These methods often require heavy computations to provide explanations. Other studies apply various deep learning reinforcement learning methods to build explanation models using various types of networks. These studies [30, 19] are commonly based on static explanation templates, result in complex models, and require parameter tuning.

Post-hoc methods are built on the assumption, that we investigate in this paper, that an explanation that makes sense to the user, even if it is not the exact reason that the recommendation was indeed issued, is acceptable to users and may have a beneficial effect for the recommendation system.

[4] suggested that providing explanations to users alongside a recommendation can help users to make more informed decisions about consuming the item. They used 3 post-hoc methods — keyword similarity, neighbors ratings, and what they call influential item computation — to explain recommendations generated by a hybrid content-based and collaborative system rating prediction system. They run a small scale user study in a books domain, attempting to understand which explanation provided the most information for the user to best understand the quality of the recommended item for her. Our paper can be seen as an extension of their preliminary work, describing a general framework for post-hoc explanations using simple methods, suggesting additional explanation types, and conducting a thorough user study in the movies domain, evaluating many more research questions.

[21] also extended the work of [4] by suggesting a different post-hoc method, applying association rule mining on the input data – the user-item rating table. The mining results with association rules, sorted by their confidence and support, that reflect links between items. Those links form the explanations that are provided to users whose input data include antecedents of the rule. The explanations, however, unlike our approach,

are limited to item-based collaboration-like statements (i.e., "item *X* is recommended because item *Y* was consumed"), and require the application of some association mining algorithm (e.g., the a-priori algorithm that the authors used [1]). Rule mining algorithms typically require heavier computations than our simple similarity-based computations. They also defined *Model Fidelity*, the portion of recommendations that can be explained. Post-hoc explanations may not always apply to all recommendations, and the goal is to provide high model fidelity.

In a gaming application, Frogger, [9] created a system that generated simple rational explanations of the agent state and actions rather than complex detailed explanations. They showed good perception of the rationales by users, further supporting our hypothesis that simple post-hoc explanations are well received by users.

The post-hoc explanation approach that we propose in this paper emphasizes simplicity, flexibility, and the ease of its application. Our method supports simple similarity based models, collaborative and content-based, as well as other simple post-hoc methods. This allows users to choose their preferred type of explanation. The main tunable parameter in our approach is the method-specific threshold for deciding which explanation is sufficiently supported to be presented to the user.

GENERATING POST-HOC EXPLANATIONS USING SIMPLE METHODS

We now present our framework for providing post-hoc explanations for complex model recommendations. The framework is presented in Figure 1.

Our method for generating recommendation along with plausible explanation operates in several stages. First, a black box recommendation model receives as input the user-item rating matrix and outputs a recommendation. Although in this paper we focus on collaborative filtering methods, this approach can be applied to other methods, such as content-based recommenders, that employ data sources other than the user-item matrix.

In the second stage, the recommended item is given as input to several explaining algorithms. In addition, each explanation algorithm receives as input additional required data sources. These explanation methods can access the data sources available to the recommender, but also other data sources as needed. For example, a possible explanation is the popularity of the item. The algorithm which produces this explanation requires data over item popularity. Another possible explanation approach is a content-based item-item method, which requires as input item content information.

The explaining algorithm is also a recommendation method, that produces a recommendation score for items, or a ranking of recommended items for the user. We use the explaining algorithm to generate such a score for the recommended item. The algorithm returns an explanation only if the recommendation score is sufficiently high. We use a method-specific threshold to decide whether the explanation is sufficiently relevant.

The explanations provided by all explaining algorithms are fed into a filter. All plausible explanations received from the explaining algorithms are filtered and one explanation is chosen to be shown to the user. For example, such a filter can be based on user preferences, or on the observed response of the user to different types of explanations. Choosing the explanation with the best score from the explanation algorithms is problematic, because these scores are not calibrated, that is, each explaining algorithm may use a different scale of scores.

USER STUDY

As we have explained above, we study the participant perception of the provided explanations. We now describe a user study applying our approach to a movie recommendation application, in which participants evaluate recommended movies, with and without explanations. The participants also provide their preferences over possible explanations for a recommended movie.

More formally, we study two hypotheses:

- Users prefer short post-hoc explanations generated by simple methods over a complete explanation of the mechanism of complex models.
- Presenting a post-hoc explanation to the user influences the user acceptance of a recommended movie.

We now explain the structure and process of the user study — the dataset and algorithms used to generate the recommendations and the explanations, and the different parts of the study. We then discuss the results that we observed.

Dataset and Algorithms

Our study is implemented in the movie recommendation domain using the Kaggle movies dataset [1], containing both ratings from MovieLens, as well as movie content data from TMDB. [2]

The dataset originally contains 45,000 movies. We filtered the dataset for two reasons — first, as we are interested in participants opinion over the presented movies, we prefer to limit our attention to relatively popular movies, to increase the likelihood that the participant is familiar with a recommended movie. Moreover, we observed that the complex models that we use provide less appropriate recommendations when the input movies have a relatively low number of user opinions. As we are not truly interested in evaluating the quality of the complex models, but rather the participant perception of the recommended movies, with and without explanations, we prefer to limit the models to items that are easier to recommend.

We hence choose to use only movies with more than 500 ratings, resulting in 3878 movies. We used all users who rated at least one of these movies, resulting in 122,147 users, and 5.7 million user-movie ratings.

For generating the recommendations, we use two complex models, an MF recommender that we implemented locally, and a variational autoencoder (VAE) [16].

[1] https://www.kaggle.com/rounakbanik/the-movies-dataset

[2] https://www.themoviedb.org/

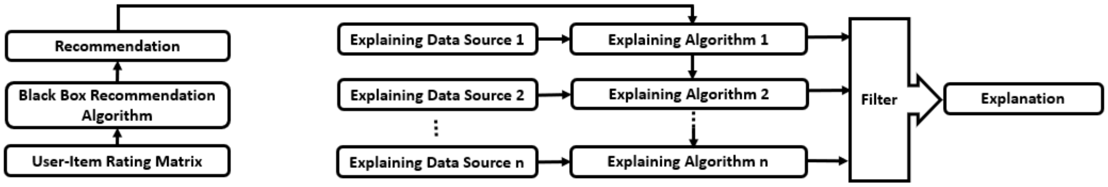


Figure 1: Generating explanation method. The method receives as input the user item-rating matrix, and additional data as input for the explanations algorithm, such as, item content,item overview, users system profile ext. The method output a recommendation and the chosen explanation.

For generating the explanations, we implemented 6 simple-to-explain algorithms. Each algorithm receives as input a user profile, and an item (i_r) that was recommended by a complex model (VAE or MF), and generates a recommendation score for that item. In addition, different algorithms take as input different data sources.

- Popularity (denoted POP in the tables below): we compute the popularity of i_r in our dataset. If the movie is sufficiently popular, we can explain the recommendation by the movie being popular. The resulting explanation reads "This movie is popular. Many users have watched it.". We set the threshold here to the 50 most popular movies.

- Item-item content based (denoted I2ICB): for each movie j that the user has rated, we compute a content similarity score between j and i_r, and take the item in the user profile with the maximal score. The content similarity is computed using the Jaccard score between the movies cast (top 5 actors only), genres and director. The resulting explanation is based on the particular content features that the items share. For example, "This movie was recommended to you because you liked j in the past, and the actor c played in both movies, and both movies are of genre g."

- User-item content-based (denoted UICB): we generate a user profile from the list of movies that the user has liked. The profile contains a score for each actor, director, and genre, based on the amount of times that a content attribute value, e.g., a specific actor, appeared in the movies that the user has liked. We then compute a weighted Jaccard score between the user profile, and i_r content attributes. The resulting explanation is based on the specific content attributes that the user profile and the item have in common. For example, such an explanation may be "This movie was recommended to you because it was directed by d, and you have liked other movies that d directed."

- Item-item overview (denoted I2ID): for each movie j that the user has rated, we compute the description similarity between j and i_r, and take the item in the user profile with the maximal score. The textual similarity between item description is computed using TF-IDF. The explanation in this case is: "This movie was recommended to you because you liked j in the past, and both movies have a similar description."

- Item-item collaborative filtering (denoted I2ICF): we compute the item-item Jaccard score, that is, the number of users who have watched both movies, divided by the union of the number of users who have watched at least one of the movies. The explanation here reads "This movie was recommended to you because you have watched movie m, and many people who like m also like this movie."

- User-user collaborative filtering (denoted U2UCF): we compute the user neighborhood using the Jaccard similarity between the sets of movies that each user has liked. Then, we compute the portion of similar users who have watched the recommended movie. This explanation reads "This movie was recommended to you because $x\%$ users who like the same movies that you did, also like this movie."

- Default explanation (denoted DEF): this is a strawman explanation that provides no additional information to the user, reading "Our system predicts that this movie is a good match for you."

For each explanation algorithm we manually tune a threshold specifying whether the explanation is sufficiently relevant to be shown to the user. We leave a smarter tuning of these thresholds to future research.

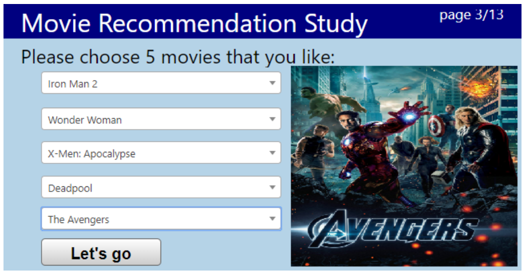
Population

We recruited to the study mostly engineering students from different academic institutes. The subjects who completed the study entered a raffle for a cash prize. Some subjects were given, in addition, a credit in an academic course. We recruited the subjects by sending an email to several mailing lists, asking people to participate in a study over recommender systems for movies.

Overall, we recruited 207 participants, 131 males, and 73 females (3 preferred not to specify gender). 24% of the participants were graduate students, 53% were undergrad students, and 23% had high school education only. 55% of the participants were 25 years old or younger, 35% were between 25-30, and 10% were above 30 years old.

Some of the participants have a background in recommendations system or related fields. 103 have taken a course in machine learning, 67 have taken a course in deep learning, 56 participants have taken a course in information retrieval, and 52 have taken a course in recommendation systems. 40% of the participants have not taken any course in those Fields.

16% reported watching a few movies each week, 46% reported watching a movie once a week, 32% once a month, and the rest (6%) almost never watch a movie Netflix is the leading movie watching channel (75%). 46% reported watching movies at the theater, 40% watch downloaded movies, and



Movie Recommendation Study

page 3/13

Please choose 5 movies that you like:

Iron Man 2

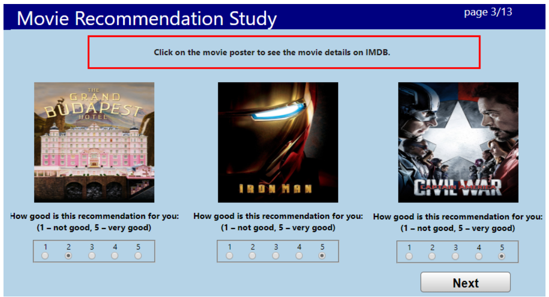
Wonder Woman

X-Men: Apocalypse

Deadpool

The Avengers

Figure 2: Choosing preferred movies. The drop-down lists on the left contain all movies in the study, and can be used to search for a movie name. Clicking on the movie poster allowed the participants to explore the movie details on the IMDB website.



Movie Recommendation Study

page 3/13

Click on the movie poster to see the movie details on IMDB.

How good is this recommendation for you: (1 – not good, 5 – very good)
1 ○ 2 ● 3 ○ 4 ○ 5 ○

How good is this recommendation for you: (1 – not good, 5 – very good)
1 ○ 2 ○ 3 ○ 4 ○ 5 ●

How good is this recommendation for you: (1 – not good, 5 – very good)
1 ○ 2 ○ 3 ○ 4 ○ 5 ●

Figure 3: Presenting two recommendations from either MF or VAE, and one popular, non-personal movie, ordered randomly. The subject must rate each recommendation.

36% watch movies on broadcast channels. We did not detect any significant difference between the various populations in the participant behavior and answers below.

We asked the participants how they decide which movie to watch. 78% use recommendations from friends, 56% read movie reviews online or in the newspaper, 25% report using some automated system to recommend movies, and 20% watch whatever is currently on. 81% are familiar with personal movie recommendations in Netflix. When asked about the quality of the Netflix recommendations , 62% reported that they sometimes liked the recommendations, 18% almost always like Netflix recommendations, 13% mostly do not like the recommendations, and 7% reported never liking these recommendations. Netflix presents some shows or movies under the title, "Because you watched X". 58% of the participants claimed that they are likely to explore recommended movies under this title, whereas 35% said that they may explore these recommendations, and 7% will not explore such a recommended movie.

Method

We now describe the process of the user study, explaining the different tasks that the test subjects performed. As we explained above, the subjects were asked to participate in a user study over movie recommendations. The invitation email, as well as the instructions at the beginning of the study, did not mention explanations. Specifically, the subjects were told that they are asked to evaluate the recommendations of a system.

Step 1: Creating a User Profile. After an instruction screen, we asked each subject to choose 5 movies which she likes (Figure 2). Using these movies, we created a CF user profile that is used as input to the black box recommendation algorithms — MF and VAE.

Once the participant clicks on the "Let's go" button, we compute two lists of 3 recommendations. For each black box algorithm, we compute two recommended movies using the provided user profile. In addition we add to each of the two recommendation lists, a randomly selected movie from the top 100 popular movies according to the IMDB popularity score. These popular movies allow us to evaluate the participant opinion over non-personalized recommendations.

Step 2: Rating Recommendations Without Explanations. During this step, the user was provided and was requested to evaluate the above two sets of recommended movies, that were presented without any explanations. The opinions of the participants over these sets serve as a baseline for the performance of the recommendation algorithms, without the influence of an explanation.

We present the recommended movies to the participant in two different screens, one containing 2 MF recommendations and one popular movie, and the other containing 2 VAE recommendations and one popular movie (Figure 3). The order of the systems, as well as the ordering within the 3 movies, is random.

Throughout the study, we avoid presenting to the subject recommended movies that were previously shown to her. If both algorithms agree on a recommended movie, we take the next movie on the recommendation list.

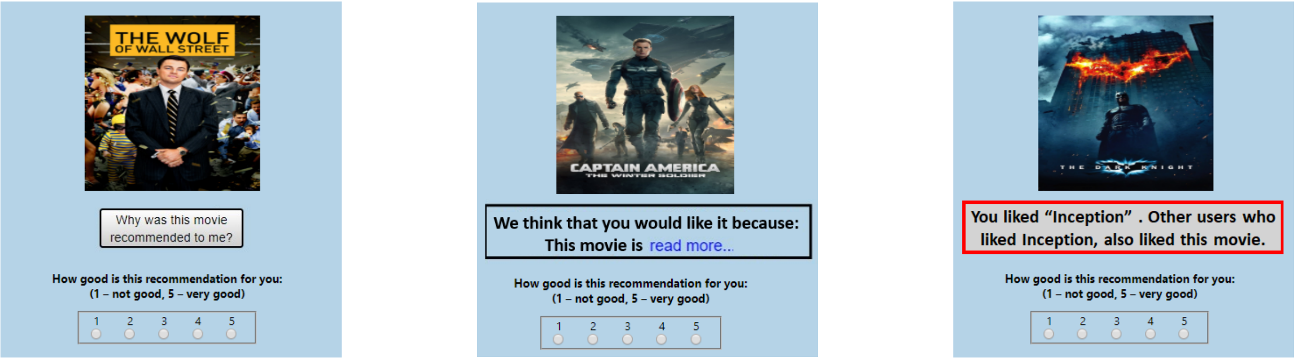
The subject rates each recommended movie in a 1-5 scale. Again, clicking on the movie poster allowed the subject to explore movie data from IMDB.

Step 3: Rating Recommendations with Explanations. We now use the black box recommenders to produce two additional recommended movies. We enrich the user profile by adding all recommended movies that the subject rated 4 or 5 in the first step, and avoid recommendations that were already presented in the first step.

In addition, we apply all the explanation generation algorithms above. We use only an explanation whose score is higher than the method-specific threshold required to be considered acceptable. From all acceptable explanations, we choose one explanation randomly. In cases where none of the algorithms returned a plausible explanation, we show a default explanation.

We used in this step 3 different methods for showing the explanations:

- Hidden: we place below the recommended movie a button saying "Why is this movie appropriate for me?". Clicking on the button opened a popup windows containing the explanation.
- Teaser: we place below the recommended movie the beginning of the explanation, followed by an ellipsis. Clicking



Why was this movie recommended to me?

How good is this recommendation for you:
(1 – not good, 5 – very good)

1 ○ 2 ○ 3 ○ 4 ○ 5 ○

(a) Hidden explanation

We think that you would like it because:
This movie is read more...

How good is this recommendation for you:
(1 – not good, 5 – very good)

1 ○ 2 ○ 3 ○ 4 ○ 5 ○

(b) Explanation teaser

You liked “Inception” . Other users who liked Inception, also liked this movie.

How good is this recommendation for you:
(1 – not good, 5 – very good)

1 ○ 2 ○ 3 ○ 4 ○ 5 ○

(c) Explicit explanation

Figure 4: Step 3: rating recommended movies with explanations. A possible simple explanation is presented for each recommendation. Each subject was shown one of the three alternative explanation displays.

on the ellipsis opened a popup window containing the explanation.

- Visible: We place below the recommended movie the explanation.

This allows us to check whether the participants are interested in an explanation, and whether they actively seek an explanation. We use a between-subjects setting here, that is, each participant was allocated to one of the 3 groups, to avoid over emphasizing the explanations due to the variations in presentation. We again ask each participant to rate 2 sets of recommendations, each containing 3 recommended movies, as in the previous step.

Step 4: Rating Explanations. In the final step of the user study we explicitly ask the subject to rate possible explanations. We again add to the user profile the successful recommendations from the previous steps, and ask for additional recommendations from the two black box algorithms, MF and VAE.

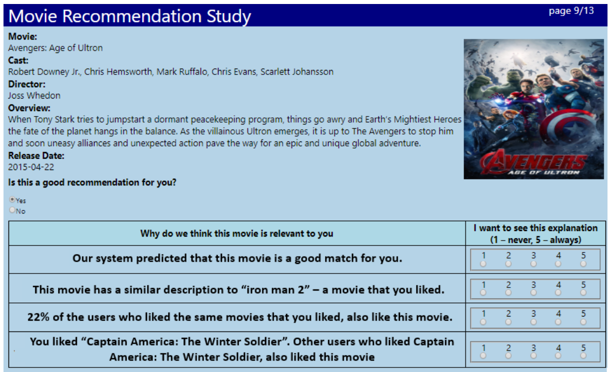
In this step, unlike the previous steps, we present to the participant a single recommended movie. In addition, we present movie content information, such as the actors, the genres, and the description, without requiring the participants to explicitly request for such details (Figure 5).

We first ask the participant to state whether she likes the recommended movie, and then present a set of explanations as to why the movie was recommended. We show all explanations that are deemed sufficiently appropriate, achieving a score higher than the method specific threshold. The participant is asked to rate each explanation on a scale of 1-5.

Each participants is shown 6 different movies in this step as well, 3 of which were generated by each of the two black box recommenders, and ordered randomly.

To summarize, we present to each participant in the steps 2-4 18 different recommendations, 7 recommendations from each of the complex models, MF and VAE, and 4 additional popular movies.

Post Study Questionnaire. After finishing step 4 above, the subject is transferred to an online questionnaire. We first ask a set of question concerning the demographic details. Then, we ask the participants questions about their movie watching habits, and their previous interactions with recommender systems. Finally, we ask questions regarding their opinions about the presented explanations.



Movie Recommendation Study page 9/13

Movie:
Avengers: Age of Ultron
Cast:
Robert Downey Jr., Chris Hemsworth, Mark Ruffalo, Chris Evans, Scarlett Johansson
Director:
Joss Whedon
Overview:
When Tony Stark tries to jumpstart a dormant peacekeeping program, things go awry and Earth's Mightiest Heroes the fate of the planet hangs in the balance. As the villainous Ultron emerges, it is up to The Avengers to stop him and soon uneasy alliances and unexpected action pave the way for an epic and unique global adventure.
Release Date:
2015-04-22

Is this a good recommendation for you?

● Yes
○ No

Why do we think this movie is relevant to you	I want to see this explanation (1 – never, 5 – always)
Our system predicted that this movie is a good match for you.	1 ○ 2 ○ 3 ○ 4 ○ 5 ○
This movie has a similar description to “iron man 2” – a movie that you liked.	1 ○ 2 ○ 3 ○ 4 ○ 5 ○
22% of the users who liked the same movies that you liked, also like this movie.	1 ○ 2 ○ 3 ○ 4 ○ 5 ○
You liked “Captain America: The Winter Soldier”. Other users who liked Captain America: The Winter Soldier, also liked this movie	1 ○ 2 ○ 3 ○ 4 ○ 5 ○

Figure 5: Step 4: rate the explanations generated by the simple models to the recommended movie of a complex model. All possible simple explanations are presented for each recommendation.

Results

We now discuss the user study results. We first review the effects of explanation on the subject perception of a recommended movie, then, we discuss subject opinion over the various explanation types. Finally, we discuss the fidelity of the various explanation methods.

Effects of Various Explanations on Movie Ratings

We now study the effect that the explanations had on the subject opinion over the recommended movies, comparing the average rating for movies without explanations and with explanations. As we explain above, in Step 3, there were 3 options for explanation presentation — hidden, requiring click a button, teaser, showing only the beginning of the explanation, and fully presenting the explanation.

Somewhat surprisingly, only for 24% of the recommendations in the first case, the participant clicked on the button, and only in 14% of the recommendations in the second case the participant clicked on the teaser. That is, in most recommendations the participants did not look at the explanations. In informal discussions following the study, participants indicated that they did not see the option to request an explanation, or did not think that they needed an explanation to decide whether the recommended movie is appropriate.

Thus, we group together here both movies in Step 2 for which no explanation was shown, and movies in Step 3 shown to participants who did not click on an explanation button or teaser. We compare this group to recommendations for which the explanation was shown. Below, when we discuss significance, we base our claims on a paired t-test.

Table 1 compares the average rating for each one of the plausible explanations, and without explanations. First, although this is not the focus of the study, the VAE method produced better recommendations than the MF method, which produced better results than recommending a random popular movie.

Looking at the explanations, we can see that the user-item content-based explanation was shown only 4 times, and is hence not statistically different than other explanations. The popularity, and the default explanations result in the lowest rating than all other explanations. That is, movies presented with either CF or CB explanations produce significantly higher ratings than the non-personal popularity explanation and the non-informative default explanation.

While the differences between the ratings can be attributed to the presented explanation, there is another plausible reason for these differences. It might be that recommended items for which a specific recommendation type applies are better recommendations. For example, it may be that when a recommended item has a strong item-item Jaccard correlation with an item in the user profile, it is considered as a better recommendation for the user, whether we explicitly tell the user about it or not.

Table 2 shows the average rating over movies that we were able to explain through one of the methods although the explanation was not shown to the subject. This occurs either in Step 2, or in Step 3 where the subject did not click on the explanation button or teaser. As can be seen, similar to the ratings in Table 1, movies for which a user based explanation exists, as well as movies with similar descriptions, receive a statistically significant (t-test p-value=0.046) higher user rating than movies for which an item-item based explanation holds. These, in turn, receive a statistically significant higher rating than movies for which only the popularity explanation holds. Finally, movies for which none of our explanation types hold, receive the lowest rating.

To conclude, on the one hand it is unclear whether the explanations that we suggest themselves truly affect the subject behavior. On the other hand, it appears that these explanations are well correlated with the way that participants perceive a recommended movie, and decide whether to rate it higher. As such, it may be that our explanations indeed capture a part of the subject decision process for her opinion over a recommended movie.

User Ratings for Explanations

As we explained above, in Step 4 we asked the participants to rate various explanations for a given recommendation. Table 3 shows the explanation ratings provided by the participants.

Somewhat surprisingly, all explanations, including the default explanation, received a relatively positive (above 3 on a 1-5 scale) rating. The only explanations that the participants significantly liked less, are the popularity explanation, and the user-item content-based explanation.

The latter is especially surprising, given that movies for which this explanation was shown, or for which this explanation holds, receive the highest user ratings in the results reported in Table 1 and Table 2. We believe that the relatively lower subject opinion for this type of explanation may be attributed not to its content, but rather to its length. As we discuss below, in the post-study questionnaire, participants reported that they prefer short explanations. This explanation is by far the longest. Note that the item-item content-based explanation may also appear to be long, in practice it is not; For the content based explanations we report all properties (actors, genres, director) that apply. A recommended movie typically has more joint property value with the user profile, containing all the movies that the user has liked (i.e., UICB), than with a single movie that the user has liked, which entails longer explanations for UICB.

Explanation Fidelity

Finally, we evaluate the explanation fidelity — the portion of recommended items for which each explanation type holds. Table 4 shows the empirical fidelity of the various explanations with respect to all recommended items in our study in the Steps 2-4. We note that the fidelity is highly sensitive to the thresholds that we set to decide which explanation is sufficiently valid to be presented. We leave an automated careful tuning of these thresholds to future research.

As can be seen, collaborative filtering fidelity is always higher than its content-based counterpart, which is not surprising, because the black box recommenders are collaborative filtering methods. Item-item explanations have higher fidelity than user-based explanations. This is not surprising, given the relatively small user profiles that we use.

It is especially interesting to look at the difference in content-based fidelity between the MF method that we use and VAE. Together with Table 2, this may explain the lower quality of recommendations computed by our MF implementation. The movies recommended by the MF method have very low content similarity to the movies that the subject has liked, and this may be the reason that participants rate them lower.

Overall, as can be seen in the bottom line of Table 2, 65% of the recommended movies could be explained by at least one of our suggested methods (except for the popularity explanation). [21] report a model fidelity of 84% at most for their created association rules. Our model fidelity is sensitive to the thresholds that we set to accept an explanation. We may be able to increase the model fidelity with more accurate and personal tuning of these thresholds.

	AE		MF		Popular		All	
	Count	Avg	Count	Avg	Count	Avg	Count	Avg
None	126	4.29	126	3.65	126	3.13	378	3.69
POP	1	5	1	2	122	3.25	124	3.26
I2ICB	25	4.52	14	4.21	-	-	39	4.41
UICB	4	4.75	-	-	-	-	4	4.75
I2ID	16	4.33	5	4	-	-	21	4.25
I2ICF	43	4.19	28	3.82	-	-	71	4.04
U2UCF	19	4.11	7	3.86	-	-	26	4.04
DEF	14	3.43	67	3.04	-	-	81	3.11

Table 1: Average ratings for movie recommendations with different explanations, and without explanations.

	Count	Avg
I2ICB	376	4.4
UICB	62	4.73
I2ID	237	4.63
I2ICF	508	4.27
U2UCF	109	4.62
Only POP	81	3.98
None applies	386	3.55

Table 2: Average rating for movies for which each recommendation type applies.

	AE		MF		All	
	Count	Avg	Count	Avg	Count	Avg
POP	46	3.35	20	3.05	66	3.26
I2ICB	98	3.71	38	3.79	136	3.74
UICB	52	3.29	4	3.75	56	3.32
I2ID	74	3.61	14	4.36	88	3.73
I2ICF	114	3.75	41	4.05	155	3.83
U2UCF	82	3.45	23	4.09	105	3.59
DEF	140	3.68	79	3.62	219	3.66

Table 3: Average user ratings for different explanations (Step 4).

Explanation	AE		MF		All	
	count	fidelity	count	fidelity	count	fidelity
POP	155	0.35	104	0.23	259	0.29
I2ICB	263	0.59	128	0.29	391	0.44
UICB	113	0.25	7	0.02	120	0.13
I2ID	194	0.43	36	0.08	230	0.26
I2ICF	312	0.7	153	0.34	465	0.52
U2UCF	181	0.4	65	0.15	246	0.28
At least one explanation by methods 2-6	365	0.82	233	0.52	579	0.65

Table 4: Model Fidelity

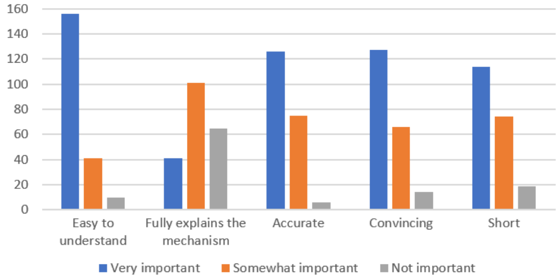


Figure 6: Explanation properties importance.

Post Study Questionnaire Results

We now discuss the participant answers to the questions concerning the explanations at the post study questionnaire. The responses below are hence biased given the explanations shown throughout the study, and may not reflect the subject opinion prior to the study.

70% of the participants reported noticing the explanations in our study, 24% noticed them only sometimes, and 6% reported not noticing the explanations at all. 60% of the participants felt that the explanations were mostly appropriate, 26% felt they were sometimes appropriate, only 1% felt that the explanations were always appropriate, and 3% felt that they were never appropriate. 71% of the participants thought that explanations can help understand the recommendation, and may influence the decision on considering the recommended item. 23%of the participants said that an explanation is interesting, but would not change their opinion over the movie. 5% responded that an explanation is not important at all, and 1% said they ignore all recommendations and hence the explanations are not relevant. Similar results were reported before for the importance of explanations in recommendation systems [12, 27].

We also asked in an open, non-obligatory question to state an explanation that they liked the best. 93 of the participants choose to answer. We categorized their free text answers into groups. 52% of the responses were related to content-based explanations. 33% preferred the collaborative filtering explanations. 10% liked the popularity explanations, and 4% liked the default explanation. [4] reports similar preference for content based explanations over CF explanations.

Figure 6 shows the participants responses about the importance of various properties of an explanation. We can see that the property that was deemed most important is that an explanation should be easy to understand. Participants also thought that an explanation should be accurate, convincing, and short. We believe that this explains the relatively low opinion of the participants concerning the content-based user-item explanation which we reported above, as this explanation is quite long.

The only property that was not deemed as important by the participants is whether the explanation fully explains the recommendation mechanism. This is in somewhat in conflict with many research attempts in the recommender system community [25, 27] that focus on providing an explanation of the way that the models operate. It appears that users, at least the participants of our study, prefer an explanation that will help them decide whether the recommended item is appropriate for them, than to understand the mechanism behind the recommendation engine.

When asked if they would like to get such recommendations in a system that they use (e.g. Netflix), 62% answered positively, 31% answered maybe and the rest (7%) answered no.

These findings, that 94% of the participants found many of our explanations to be appropriate, and that most people would have liked to see such explanations in a system that they use, together with the relatively low importance of revealing the recommendation engine behavior, further support our intuition, that post-hoc explanations generated by simple methods can provide valuable information that users appreciate.

CONCLUSION

In this paper we suggest a simple method for generating post-hoc explanations for recommendations generated by complex, difficult to explain, models. We use a set of easy to explain recommendation algorithms, and when their output agrees with the recommendation of the complex model, consider the explanation of the simple model as a valid explanation for the recommended item. While these explanations are clearly not transparent, we argue that they provide valuable information for the users in making decisions concerning the recommended items.

We study two research questions. First, whether users prefer our simple post-hoc explanations to explanations of the mechanism of the neural network or the matrix factorization model. Indeed, in our post study questionnaire, users stated that it is more important for an explanation be short and clear, than to fully explain the algorithm.

Second, we checked whether presenting a post-hoc explanation influences the behavior of users. For some of our explanations, namely, the I2ICB explanation and the I2ID explanation, the average rating was higher when an explanation was presented than the average rating when no explanation was presented. For other explanations, this did not hold. We speculate that this was due to the explanation length and complexity. Perhaps a future, simpler phrasing of the explanation would lead to more pronounced effects.

To support our claims, we use a user study in the movie domain, showing that some explanations may affect the user opinion over the recommended item. We also show that movies that can be explained by our method may be better items to recommend. We evaluate subject opinion over the different explanations that we suggest, showing that participants preferred item-item explanations to user-based explanations. The subjects also stated that it is more important for an explanation to be easy to understand, convincing, and short, than to uncover the underlying operation of the recommendation engine.

Our method can be easily extended by using additional explainable recommenders. In the future we will apply more methods. We will also study methods for automatically selecting a method-specific threshold for deciding if an explanation is valid, instead of the manually tuned threshold that we currently use.

REFERENCES

[1] Rakesh Agrawal, Ramakrishnan Srikant, and others. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, Vol. 1215. 487–499.

[2] Fernando Amat, Ashok Chandrashekar, Tony Jebara, and Justin Basilico. 2018. Artwork personalization at netflix. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 487–488.

[3] Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[4] Mustafa Bilgic and Raymond J Mooney. 2005. Explaining recommendations: Satisfaction vs promotion. In *Beyond Personalization Workshop, IUI*, Vol. 5. 153.

[5] John S Breese, David Heckerman, and Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 43–52.

[6] 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*. 909–912.

[7] Weiyu Cheng, Yanyan Shen, Linpeng Huang, and Yanmin Zhu. 2019. Incorporating Interpretability into Latent Factor Models via Fast Influence Analysis. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 885–893.

[8] Hendrik Drachsler, Katrien Verbert, Olga C Santos, and Nikos Manouselis. 2015. Panorama of recommender systems to support learning. In *Recommender systems handbook*. Springer, 421–451.

[9] Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O Riedl. 2019. Automated rationale generation: a technique for explainable AI and its effects on human perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 263–274.

[10] Michael D Ekstrand, John T Riedl, Joseph A Konstan, and others. 2011. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction* 4, 2 (2011), 81–173.

[11] Bruce Ferwerda, Kevin Swelsen, and Emily Yang. 2018. Explaining Content-Based Recommendations. *New York* (2018), 1–24.

[12] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. 241–250.

[13] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

[14] Yehuda Koren. 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 4, 1 (2010), 1.

[15] Yehuda Koren and Robert Bell. 2015. Advances in collaborative filtering. In *Recommender systems handbook*. Springer, 77–118.

[16] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*. 689–698.

[17] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.

[18] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. 2015. Recommender system application developments: a survey. *Decision Support Systems* 74 (2015), 12–32.

[19] 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 (RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 31–39. DOI:http://dx.doi.org/10.1145/3240323.3240354

[20] Xia Ning, Christian Desrosiers, and George Karypis. 2015. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*. Springer, 37–76.

[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*. 2060–2069.

[22] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook*. 1–34.

[23] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[24] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender systems handbook*. Springer, 257–297.

[25] Rashmi Sinha and Kirsten Swearingen. 2002. The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems*. 830–831.

[26] Brent Smith and Greg Linden. 2017. Two decades of recommender systems at amazon. com. *Ieee internet computing* 21, 3 (2017), 12–18.

[27] Nava Tintarev and Judith Masthoff. 2007. A survey of explanations in recommender systems. In *2007 IEEE 23rd international conference on data engineering workshop*. IEEE, 801–810.

[28] Nava Tintarev and Judith Masthoff. 2015. Explaining Recommendations: Design and Evaluation. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 353–382. DOI:http://dx.doi.org/10.1007/978-1-4899-7637-6_10

[29] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1235–1244.

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

[31] Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192* (2018).

[32] 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*. 83–92.