Contrastive Explanations for Recommendation Systems

Anonymous CogSci submission

Abstract

Recommendation systems are widely used and are present in many applications, such as movie recommendations, product sales, and content providers. However, current recommendation systems are usually stern and lack the ability to explain their decisions or allow the users to question them.

In this paper, we develop an automatic method that, given a contrastive query from the user, generates contrastive explanations based on items' features and users' preferences (provided as ratings). That is, once receiving a recommendation, the users have the option to ask the system why it did not recommend a specific different item. Our method enables a recommendation system to reply with a meaningful and convincing personalized explanation. For example, the recommendation system may recommend the user to buy a Samsung S22 phone. The user may ask the system why it did not recommend the Xiaomi 12. Based on the user's preferences, all other users' preferences, and the specific phones in question, our method might infer that a good camera is particularly important to the user, and thus, say that the Samsung S22 includes a better camera than the Xiaomi 12.

We compose a new dataset based on user ratings of the most popular cell phones in the US in 2022. Based on this dataset, we run an experiment with 100 human participants who are recommended an item and shown contrastive explanations generated by our method, as well as two additional baseline methods. We show that humans are more convinced that the recommended item is better than the contrastive item when using our contrastive explanations.

1 Introduction

Recommendation systems are becoming significant in extensive aspects of our daily lives. In some areas, such as medicine (Suphavilai, Bertrand, & Nagarajan, 2018) and finance (Zibriczky, 2016), recommendation systems are used as tools that help making decisions at high stakes. In these settings, the system is expected to justify its recommendations, since the cost of a wrong decision may be high. However, state-of-art techniques used in recommendation systems are becoming more and more inscrutable due to the complexity of models and high dimension of data on which the models are trained. Arguably, the most significant means to increase trust and transparency in recommendation systems is by providing explanations for the system's decisions that are concise and understandable to humans (Tintarev & Masthoff, 2015).

In order to explain the outputs of a recommendation system, two strategies that are common in the field of explainable AI can be applied. The first is to adopt interpretable models, whose decision mechanism is transparent, and thus, we can naturally provide explanations for the model decisions (Abdollahi & Nasraoui, 2017; Bauman, Liu, & Tuzhilin, 2017). Explanations of this type are called ante-hoc, and they seek to understand the inner workings of a model while the model is in the process of making decisions. However, they often suffer from lower prediction accuracy, which is known as the trade-off between complexity and accuracy (Doshi-Velez & Kim, 2017). In addition, when recommendations are provided as an external service, the inner workings of the system are the intellectual property of the service provider and cannot be revealed. The second strategy, which does not suffer from these limitations, is to provide post-hoc explanations, i.e., after the recommendation has been given. Post-hoc explanations do not precisely reflect the underlying recommendation model. Instead, they present rationale, plausible, and valuable information to the user. Consequently, post-hoc explanations are independent of the main recommendation algorithm, and therefore they have the potential to be used across a greater diversity of recommendation systems.

Indeed, what constitutes good post-hoc explanations has been extensively studied in social science, cognitive science, and psychology (Nozick, 1983; Lombrozo, 2006). In particular, Lipton (1990) investigated different types of explanations, and concluded that they should be contrastive. That is, instead of explaining why an event happened, explanations are more convincing if the person receiving the explanation can ask why one event happened rather than another specific event. A contrastive explanation in the context of recommendation systems should allow the user to pose questions of the form "why did you recommend this item rather than another specific item?". To the best of our knowledge, currently, no recommendation system allows users to pose contrastive queries. Therefore, in this work we develop a method that generates contrastive explanations for recommendation systems. With our method the system can reply with a meaningful and convincing personalized explanation which is based on the features of the recommended item, the contrasting item, the specific user's preferences, and all users' preferences.

Specifically, given a user who was recommended an item p, the user may ask the system why it did not recommend another item q. The essence of our method is to select a set of features and show the user the differences between p and q in the selected features. For example, suppose that item p is a Galaxy S22 cell phone, item q is an iPhone 13 cell phone,

and the selected set of features is price and main camera resolution. Our method will generate the explanation:

> *"The recommended cell phone Galaxy S22 costs* \$528*, compared to iPhone 13, which costs* \$699*; The recommended cell phone Galaxy S22 main camera's resolution is* 50 *MP, compared to a resolution of* 12 *MP in the iPhone 13"*

In order to select an appropriate set of features, we develop the Features Selector for Contrastive eXplanations (FS-CX) algorithm. FS-CX treats each item as a point in the space spanned by all the features, and samples hypothetical items that are close to p, q, and in the area between them. The hypothetical items are evaluated by the recommendation system, which outputs their utilities. The hypothetical items and their utilities are then used to train a linear regression model, from which we derive weights that represent the importance of each feature, tailored to the area between p and q. These weights are multiplied by the difference between p and q to obtain each feature's influence. FS-CX then iterates over the features in descending order of influence, and selects the first set of features such that if we replace their values in q with their values in p, the recommendation system will evaluate the utility of q equal or higher than the utility of p.

In order to evaluate the performance of FS-CX, we first composed a dataset of 34 recent cell phones and their features. We recruited 100 human participants who were asked to rate the cell phones. Then, we recruited an additional set of 100 human participants for our main survey. Each participant in the survey was first requested to rate some cell phones, and then presented with the cell phone recommended by the recommendation system along with another cell phone. The participant was shown contrastive explanations, attempting to convince the participant that the cell phone recommended by the recommendation system is better; these explanations were generated by FS-CX and other baselines. Finally, the participants were requested to rate how convincing are each of the explanations. We show that the explanations that were generated by FS-CX achieved higher average rating compared to the other explanations.

To summarize, the main contribution of this paper is that it provides a method that can generate more human-like explanation to recommendation systems and can be applied to any feature-based recommendation system models. More specifically, we propose a method that generates post-hoc contrastive explanation for recommendation systems. The dataset created in this paper is of independent interest, as it is a feature-based dataset with many numerical values.

2 Related work

Our work is related to the field of Explainable AI (XAI) (Core et al., 2006; Gunning et al., 2019). In a typical XAI setting, the goal is to explain the output of an AI system to a human. This explanation is important for allowing the human to trust the system, better understand it, and to allow transparency of the system's output (Adadi & Berrada, 2018). Other XAI systems are designed to provide explanations, comprehensible by humans, for legal or ethical reasons (Doran, Schulz, & Besold, 2017). For example, an AI system for the medical domain might be required to explain its choice for recommending the prescription of a specific drug (Holzinger, Biemann, Pattichis, & Kell, 2017).

One common type of explanation in recommendation systems provides explanations that answer the question "Why a specific item was recommended?" There are many ways for answering such a question. Specifically, Herlocker et al. (2000) propose a user-style explanation. That is, the system presents the user a group of "neighborhood" users, who have similar interests to hers, and these neighborhood users provided high ratings for the recommended item. Quijano-Sanchez et al. (2017) propose a social-based explanation. That is, the system presents the user with a list of her friends who also liked the recommended item. Vig et al. (2009) adopt movie tags as features to generate recommendations and explanations. To explain the recommended movie, the system displays the movie features and tells the user why each feature is relevant to her. Hou et al. (2019) use radar charts to explain why an item is recommended.Peake and Wang (2018) present an approach that extracts explanations from latent factor-based recommendation systems, by training association rules on the output of a matrix factorization black-box model. Nóbrega and Marinho (2019) introduce LIME-RS, which was later improved by Chanson et al. (2021). LIME-RS is an adaptation of LIME to recommendation systems. LIME is a popular approach for explaining a machine learning model's output by identifying the top-n features that have the greatest impact on the model's output (Ribeiro, Singh, & Guestrin, 2016). SHAP is another common approach for explaining a machine learning model's output; it relies on computing the average, over all permutations, of the marginal contributions of each feature (Lundberg & Lee, 2017). Indeed, Guo et al. (2021) adapt SHAP to derive explanations for recommendation systems. Shmaryahu et al. (2020) use a set of simple, easily explainable recommendation algorithms to provide explanations for a complex recommendation system. Their method attempts to find a simple explainable recommendation system that agrees with the complex model on a recommended item and applies the explanation of the simple model.

Another type of explanation is counterfactual explanations, which answer the question "Why a specific item was recommended rather than *any* other item?" Counterfactual explanations consider changes to features and events that alter the outputs of the recommendation system (Wachter, Mittelstadt, & Russell, 2017). A typical counterfactual explanation describes a causal situation: "If the specific event did not occur, a different item would have been recommended." Unlike methods that try to approximate the original models, counterfactual explanations examine changes in the output of the original model and therefore have high fidelity (Kaffes,

Sacharidis, & Giannopoulos, 2021). One example for a method providing counterfactual explanations is PRINCE, introduced by Ghazimatin et al. (2020). PRINCE provides an explanation by finding a set of minimal actions performed by the user that, if removed, changes the recommendation to a different item based on random walks over dynamic graphs. This approach was later improved by Kaffes et al. (2021), who used the normalized length and the importance of a candidate to guide the search. Zhong and Negre (2022) develop a method for providing counterfactual explanations that sorts all features according to their SHAP values, and selects the first set of features that, if their value is changed randomly, the recommendation system will change its recommendation.

In this work, we focus on a third type of explanations, which are contrastive explanations. A contrastive explanation in the context of recommendation systems answers the question "Why a specific item was recommended rather than another **specific** item?" To the best of our knowledge, currently, no recommendation system allows users to pose contrastive queries.

Our work should not be confused with interactive recommendation systems (He, Parra, & Verbert, 2016; Gao, Lei, He, de Rijke, & Chua, 2021), a field that focuses on updating the system-model based on interaction with the user and does not consider interactive explanations as we do.

3 FS-CX

In this section, we represent our Features Selector for Contrastive eXplanations (FS-CX) algorithm. FS-CX builds upon a utility based recommendation system, i.e., a system that, given a user, evaluates the utility of each available item for the user and recommends the item with the highest utility. FS-CX also requires that the recommendation system uses a set of item features F, for example the item's brand or price; the features can be numeric or categorical. The recommendation system may also use features of the user (e.g., demographic information) and additional data (e.g., collaborative-based).

FS-CX is used for generating contrastive explanations. That is, assuming a user is given a recommendation for item p, she may ask the system why it did not recommend another specific item q. The system must respond with a compelling explanation that is based on the user's preferences. To create an explanation, FS-CX selects a set of features that best explain how p is better than q.

FS-CX is formally described by Algorithm 1. Intuitively, FS-CX generates a set of hypothetical items, X, by independently sampling each feature from a uniform distribution between p's value for that feature and q's value for it. That is, for every item $x \in X$, each feature $f \in F$ of x, denoted by x_f, is independently sampled from $U(\min\{p_f, q_f\}, \max\{p_f, q_f\})$. Figure 1 illustrates the sampling method used by FS-CX for generating hypothetical items (when there are only 2-features). As depicted by the figure, the feature values are between those of p and those of q; therefore, the hypothetical items should capture the variations between p and q considering all features.

Algorithm 1: FS-CX algorithm

Input : Recommendation system R; Set of item features F; User u; Recommended item p; Contrastive item q; Number of samples n.
Output: A set of explanatory features.

$X, E \leftarrow \emptyset$
for $i \leftarrow 1$ *to* n **do**
  for $f \in F$ **do**
    $x_f \leftarrow$ sample from $U(\min\{p_f, q_f\}, \max\{p_f, q_f\})$
  $X \leftarrow X \cup x$
$Y \leftarrow R(X, u)$
train a linear regression model, T, on (X, Y)
$W \leftarrow T$'s weights multiplied by $(p - q)$
for $f \in F$ *sorted in descending order according to* W **do**
  $E \leftarrow E \cup f$
  $q_f \leftarrow p_f$
  if $R(q, u) \geq R(p, u)$ **then**
    break
return E

Once the hypothetical items X are generated, they are fed to the recommendation system for evaluating their utility. The hypothetical items along with their utilities are used to train a linear regression model. FS-CX uses the model to derive weights, which their absolute values represent the importance of a unit of each feature in the area between p and q. FS-CX multiplies these weights, for each feature f, by the distance $p_f - q_f$. This value represents the explanatory power of each feature.

FS-CX then iterates over the features, ordered by their explanatory powers (in descending order). FS-CX selects the first set of features such that if we replace their values in q to their values in p, the new item will obtain a utility that is higher or equal to that of p. Intuitively, this approach finds the disadvantages of q such that if they are addressed (using the values of the features in p), the recommendation system will evaluate the utility of q equal or higher than the utility of p. FS-CX limits the number of selected features by a constant, k. As mentioned earlier, the explanation is generated from this set of features.

4 Experimental design

In order to evaluate our method, FS-CX requires a dataset with item features and user ratings. Unfortunately, many of the available rating datasets focus on content recommendations (e.g., movies and books), and thus they include very few numerical or categorical features. That is, these datasets are less suitable for our setting. Therefore, we composed a cell phone dataset with item features that includes user ratings. To that end, we first collected data on the most popular

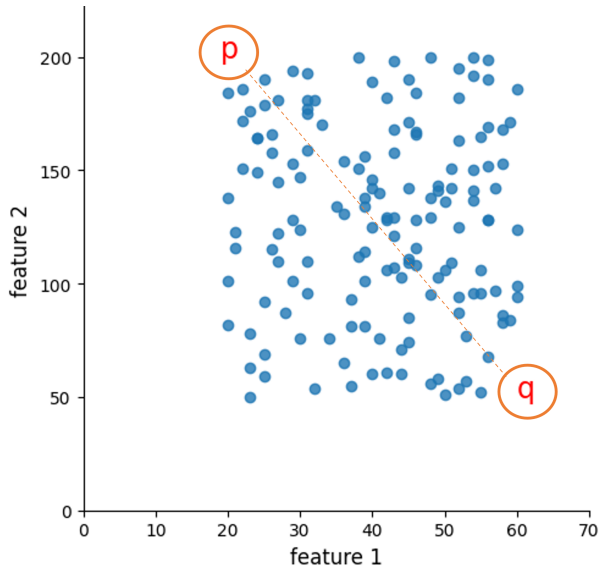

Figure 1: Illustration of the sampling method for generating hypothetical items with 2 features.

cell phones in the US in 2022[1]. The data for each cell phone consists of the following features: brand, model, operating system, memory size (in gigabytes), performance rating (AnTuTu benchmark), camera resolution (in megapixels), battery size (in mAh), screen size (in inches), weight (in grams), price (in US dollars), and release date. We collect the price of each cell phone from Amazon and Best-Buy (accessed on Aug 2022). Overall, in our dataset there are 34 cell phones with 13 features.

In order to elicit the ratings, we conducted a survey on Mechanical Turk (Paolacci, Chandler, & Ipeirotis, 2010). Each participant in the survey was presented with 10 random cell phones and was asked to indicate how likely she is to purchase each of the cell phones at the given price, on a scale from 1 (very unlikely) to 10 (very likely).

To encourage the participants to provide more meaningful ratings, we required them to add an explanation for each rating. This also allowed us to understand the logic behind the participants' ratings and verify that they are aligned with their true preferences. For example, one participant rated all the iPhones presented to him as 10, and all other cell phones as 1. He explained that "Apple products are always promising" and "iPhone's are such great products that work with everything". On his rating for the other cell phones he explained that "Androids are poorly made" and "I will never use an Android". Another participant rated all cell phones whose price is over $800 as 1, and explained that the price is "too expensive", while cell phones whose price is around $600 and have good specification he rated as 8 or 9. Figure 2 shows a screenshot from the rating elicitation survey. In addition, we asked each participant to provide personal information: age, gender and occupation.

We recruited 100 participants, 51 of them were males, 46 females, and 3 participants who chose not to specify their gender. The average age of the participants was 36.39. Since each participant rated 10 cell phones, this survey resulted with a total of 1000 ratings. We note that the average rating was 6.7, with the Apple iPhone 13 and iPhone 13 pro achieving the highest average rating of 8.0, and Motorola Moto G Play achieving the lowest average rating of 5.1. Apple iPhone 13 also has the lowest standard deviation (1.6), while Xiaomi 12 Pro has the highest (3.4).

Recall that FS-CX builds upon a utility based recommendation system. Therefore, we use the following two state-of-the-art recommendation systems.

- *MeLU* (Lee, Im, Jang, Cho, & Chung, 2019), a meta-learning-based recommendation system that can provide a recommendation based on only a few ratings from each user.

- *Wide & Deep* (Cheng et al., 2016), a recommendation system that combines the strengths of both linear (i.e., wide) models and deep learning models. A linear model with a wide set of features can memorize the interaction between the features, while a deep neural network can generalize better to unseen feature combinations.

In order to evaluate the explanations generated by our method, we ran a survey on Mechanical Turk. The participants were asked to provide demographic information and to rate 10 cell phones. Then, two cell phones were presented to each participant. The first, is the cell phone recommended by a recommendation system (for the specific participant); we denote this cell phone by p. The second cell phone, which we denote by q, was chosen at random from the cell phones the participant did not rate. The participant was told that q was preferred by a user with similar preferences. We then presented three explanations that describe, based on the features of the two cell phones, why p is better for the participant than q. The explanations were based on features selected by the following algorithms:

- **FS-CX**: As described in Section 3.

- **Linear Regression**: This algorithm first trains a linear regression model on the items rated by the participant. The algorithm then selects the features with the highest absolute weights.

- **Random**: A random selection of features, conditioned on their values being different between p and q.

Since the number of features selected by FS-CX varies, and in order to eliminate any dependence on the number of features selected, the other two methods selected the same number of features as FS-CX. The order in which the explanations (each

[1] https://www.theverge.com/22163811/best-phone
https://www.techadvisor.com/article/724318/best-smartphone.html
https://www.tomsguide.com/best-picks/best-phones

brand	model	operating system	internal memory	RAM	performance	main camera	selfie camera	battery size	screen size	weight	price	release date	rating	your explanation
Sony	Xperia Pro	Android	512GB	12GB	6.82	12 MP	8 MP	4000 mAh	6.5"	225 g	$1998	27/01/2021		
Samsung	Galaxy S22 Plus	Android	128GB	8GB	7.22	50 MP	10 MP	4500 mAh	6.6"	195 g	$899	25/02/2022		
OnePlus	Nord 2T	Android	128GB	8GB	6.04	50 MP	32 MP	4500 mAh	6.43"	190 g	$379	21/05/2022		
Google	Pixel 6	Android	128GB	8GB	6.76	50 MP	8 MP	4614 mAh	6.4"	207 g	$499	28/10/2021		
Motorola	Moto G Play (2021)	Android	32GB	3GB	1.42	13 MP	5 MP	5000 mAh	6.5"	204 g	$159	14/01/2021		
Oppo	Find X5 Pro	Android	256GB	12GB	10.12	50 MP	32 MP	5000 mAh	6.7"	218 g	$987	14/03/2022		
Samsung	Galaxy S22 Ultra	Android	128GB	8GB	9.68	108 MP	40 MP	5000 mAh	6.8"	228 g	$840	25/02/2022		
Samsung	Galaxy A32	Android	64GB	4GB	2.20	48 MP	13 MP	5000 mAh	6.5"	205 g	$199	22/01/2021		
Apple	iPhone XR	iOS	64GB	3GB	4.22	12 MP	7 MP	2942 mAh	6.1"	194 g	$236	26/10/2018		
Google	Pixel 6a	Android	128GB	6GB	6.88	12 MP	8 MP	4410 mAh	6.1"	178 g	$449	21/07/2021		

Figure 2: Screenshot from the rating elicitation survey.

by a different algorithm) were presented to the participant was random. Clearly, the participants were not aware of how each explanation was generated.

Each participant was asked to indicate to what extent each explanation convinced her that p is better for her than q. The participant could select a score between 1 and 7 using a Likert scale (Joshi, Kale, Chandel, & Pal, 2015).

For evaluating our approach we recruited 100 participants, half received a recommendation according to MeLU and the other half according to Wide & Deep. Of the 100 participants 47 were males and 53 females. The average age of the participants was 37.69. We set a requirement on Mechanical Turk that the approval rate of the workers must be at least 99% and did not require the Turkers to be masters.

5 Results

Figures 3 and 4 compare the performance, in terms of average score, of the explanations generated by our method, using the FS-CX algorithm and the other algorithms, in both MeLU and Wide & Deep recommendation systems. As depicted by the figures, the explanations that were generated by our method outperformed all other explanations. These differences are statistically significant ($p < 0.01$; using a student t-test). That is, the human participants, when presented with explanations according to our method, are more convinced that the recommended item (p) is better for them than the contrastive item (q) than when using contrastive explanations generated by other methods. Note that our approach works well, and provides similar results, with both recommendation systems.

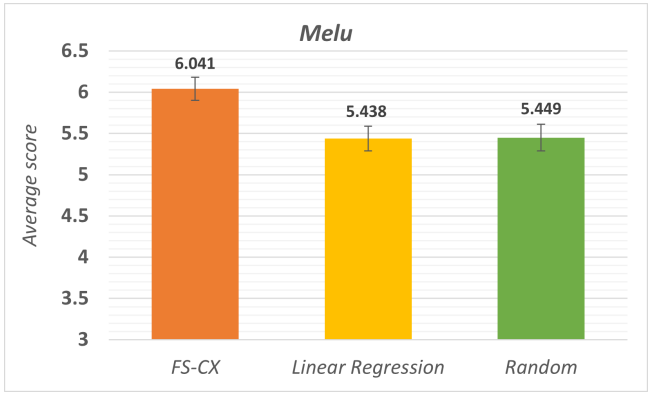

Figure 3: Average score for MeLU for every type of explanation. Error bars present the standard error.

Figure 4: Average score for Wide & Deep for every type of explanation. Error bars present the standard error.

We conclude this section by providing two examples of the explanations generated by our approach. For one participant, the recommended cell phone was Samsung Galaxy S22 Ultra and the contrastive cell phone was Apple iPhone 13 Pro Max. Our method generated the following explanation:

> *"The recommended cell phone Galaxy S22 Ultra RAM memory is 8GB, compared to a memory of 6GB in the iPhone 13 Pro Max. The recommended cell phone Galaxy S22 Ultra selfie camera resolution is 40 MP, compared to a resolution of 12 MP in the iPhone 13 Pro Max. The recommended cell phone Galaxy S22 Ultra brand is Samsung, compared to Apple, which is the brand of the iPhone 13 Pro Max."*

Another participant was recommended Samsung Galaxy A53 and the contrastive cell phone was Xiaomi 12 Pro. Our method generated the following explanation.

> *"The recommended cellphone Galaxy A53 brand is Samsung, compared to Xiaomi, which is the brand of the 12 Pro. The recommended cellphone Galaxy A53 costs* $312*, compared to a price of* $618 *in the 12 Pro. The recommended cellphone Galaxy A53 battery size is 5000 mAh, compared to a battery size of 4600 mAh in the 12 Pro."*

6 Discussion

In our approach we sample new hypothetical items that are close (in terms of their feature values) to the two items being compared, and the virtual space between them. For this reason, our method can only be applied to recommendation systems that can provide a utility to the hypothetical items. Indeed, content based recommendation systems, which base their recommendation on the feature values of the items, are a natural candidate that our method can be applied to.

However, since the hypothetical items do not have any ratings from users, our method cannot be directly applied to a pure collaborative filtering based recommendation system, which requires ratings for each item in order to provide a recommendation. This is somewhat unfortunate, as collaborative filtering based recommendation systems can help users discover new interests, and although the system might not know the user's interest in a given item, the recommendation system might still recommend it since similar users are interested in that item. On the other hand, a content-based model can only make recommendations based on the existing interests of the user, and hence, the model only has limited ability to expand on the users' existing interests. Collaborative filtering-based recommendation systems have also been shown to perform better than content-based recommendation systems (Thorat, Goudar, & Barve, 2015). Nevertheless, we note that the requirement to consider items with no ratings from users is not specific to our setting, but is a common issue known as the cold-start problem (Su & Khoshgoftaar, 2009; Lam, Vu, Le, & Duong, 2008). Therefore, there exist multiple hybrid approaches that intend to combine the advantages of both types of recommendation systems; indeed, the two recommendation systems that we have considered in our work (MeLU and Wide & Deep) are considered hybrid recommendation systems.

7 Conclusion

In this paper, we developed a method that, given a contrastive query from the user, generates contrastive explanations based on items' features and users' preferences. We showed that when applying our method, humans are more convinced that the recommended item is better for them, than when using contrastive explanations generated by other methods. To the best of our knowledge, this is the first work that enables contrastive queries to recommendation systems. Embedding our method in a recommendation system will allow it to be more attentive and not only provide a recommendation, but also allow the user to interact with it and question its decision.

Our method can also be applied for comparing two items, neither of which were recommended by a recommendation system. Item comparison is a widely used feature, and many e-commerce websites provide it. In order to apply our method, one must initially consider the first item as p and the second item as q, which will provide the advantages that the first item has over the second, and then consider the first item as q and the second item as p, which will provide the advantages that the second item has over the first.

8 Future Work

In future work, we intend to allow the user to further ask why the system did not recommend another, different item, q'. A naive approach, which can be used as a baseline, is to use our approach from the first phase and find features that explain the advantages of p over q'. However, we hypothesize that an algorithm that considers also the previously queried item q along with the previously provided explanation, will result in better explanations for the user. Clearly, the interaction between the user and the system may include several contrastive queries, and we intend to develop algorithms that consider *all* previously contrastive queries and previously provided explanations to generate a new explanation.

Another setting that we intend to tackle in the future is to allow the user to post multiple contrastive queries simultaneously. Our algorithm should provide a more holistic explanation, accounting for all the contrastive queried items together.

References

Abdollahi, B., & Nasraoui, O. (2017). Using explainability for constrained matrix factorization. In *Proceedings of the eleventh acm conference on recommender systems* (p. 79–83).

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, *6*, 52138-52160.

Bauman, K., Liu, B., & Tuzhilin, A. (2017). Aspect based recommendations: Recommending items with the most

valuable aspects based on user reviews. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (p. 717-–725).

Chanson, A., Labroche, N., & Verdeaux, W. (2021). Towards local post-hoc recommender systems explanations. In *Proceedings of the 23rd international workshop on design, optimization, languages and analytical processing of big data (dolap)*.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., ... Shah, H. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* (p. 7—10).

Core, M. G., Lane, H. C., Van Lent, M., Gomboc, D., Solomon, S., & Rosenberg, M. (2006). Building explainable artificial intelligence systems. In *Aaai* (pp. 1766–1773).

Doran, D., Schulz, S., & Besold, T. R. (2017). What does explainable AI really mean? a new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794*.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*.

Gao, C., Lei, W., He, X., de Rijke, M., & Chua, T.-S. (2021). Advances and challenges in conversational recommender systems: A survey. *AI Open*, *2*, 100-126.

Ghazimatin, A., Balalau, O., Saha Roy, R., & Weikum, G. (2020). Prince: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proceedings of the 13th international conference on web search and data mining* (pp. 196–204).

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., & Yang, G.-Z. (2019). Xai—explainable artificial intelligence. *Science robotics*, *4*(37).

Guo, M., Yan, N., Cui, X., Hughes, S., & Al Jadda, K. (2021). Online product feature recommendations with interpretable machine learning. *arXiv preprint arXiv:2105.00867*.

He, C., Parra, D., & Verbert, K. (2016). Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, *56*, 9–27.

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 acm conference on computer supported cooperative work* (pp. 241–250).

Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain? *arXiv preprint arXiv:1712.09923*.

Hou, Y., Yang, N., Wu, Y., & Yu, P. S. (2019). Explainable recommendation with fusion of aspect information. *World Wide Web*, *22*(1), 221–240.

Joshi, A., Kale, S., Chandel, S., & Pal, D. K. (2015). Likert scale: Explored and explained. *British Journal of Applied Science & Technology*, *7*(4).

Kaffes, V., Sacharidis, D., & Giannopoulos, G. (2021). Model-agnostic counterfactual explanations of recommendations. In *Proceedings of the 29th acm conference on user modeling, adaptation and personalization* (pp. 280–285).

Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008). Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on ubiquitous information management and communication* (p. 208–211).

Lee, H., Im, J., Jang, S., Cho, H., & Chung, S. (2019). Melu: Meta-learned user preference estimator for cold-start recommendation. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplement*, *27*, 247—266.

Lombrozo, T. (2006). The structure and function of explanations. *Trends in Cognitive Sciences*, *10*, 464–470.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems* (p. 4768-–4777).

Nóbrega, C., & Marinho, L. (2019). Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th acm/sigapp symposium on applied computing* (pp. 1671–1678).

Nozick, R. (1983). *Philosophical explanations*. Harvard University Press.

Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on amazon mechanical turk. *Judgment and Decision making*, *5*(5), 411–419.

Peake, G., & Wang, J. (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* (pp. 2060–2069).

Quijano-Sanchez, L., Sauer, C., Recio-Garcia, J. A., & Diaz-Agudo, B. (2017). Make it personal: a social explanation system applied to group recommendations. *Expert Systems with Applications*, *76*, 36–48.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (p. 1135—1144).

Shmaryahu, D., Shani, G., & Shapira, B. (2020). Post-hoc explanations for complex model recommendations using simple methods. In *Proceedings of the 7th joint workshop on interfaces and human decision making for recommender systems* (pp. 26–36).

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*.

Suphavilai, C., Bertrand, D., & Nagarajan, N. (2018). Predicting cancer drug response using a recommender system. *Bioinformatics*, *34*, 3907–3914.

Thorat, P. B., Goudar, R. M., & Barve, S. (2015). Sur-

vey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, *110*(4), 31–36.

Tintarev, N., & Masthoff, J. (2015). Explaining recommendations: Design and evaluation. In *Recommender systems handbook* (pp. 353–382). Springer.

Vig, J., Sen, S., & Riedl, J. (2009). Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on intelligent user interfaces* (pp. 47–56).

Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard journal of law & technology*, *31*, 841–887.

Zhong, J., & Negre, E. (2022). Shap-enhanced counterfactual explanations for recommendations. In *Proceedings of the 37th acm/sigapp symposium on applied computing* (p. 1365—1372). Association for Computing Machinery.

Zibriczky, D. (2016). Recommender systems meet finance: a literature review. In *Finrec*.