Post-Hoc Attribute-Based Explanations for Recommender Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recommender systems are ubiquitous in most of our interactions in the current digital world. Whether shopping for clothes, scrolling YouTube for exciting videos, or searching for restaurants in a new city, the recommender systems at the back-end power these services. Most large-scale recommender systems are huge models trained on extensive datasets and are black-boxes to both their developers and end-users [47, 48]. Prior research has shown that providing recommendations along with their reason enhances trust, scrutability, and persuasiveness of the recommender systems [35, 8, 31]. Recent literature in explainability has been inundated with works proposing several algorithms to this end [47]. Most of these works provide item-style explanations, i.e., 'We recommend item A because you bought item B.' We propose a novel approach to generate more fine-grained explanations based on the user's preference over the attributes of the recommended items [18]. We perform experiments using real-world datasets and demonstrate the efficacy of our technique in capturing users' preferences and using them to explain recommendations. We also propose ten new evaluation metrics and compare our approach to six baseline methods.

1 Introduction

Recommender systems drive the modern discovery of subjects of interest. Examples are ubiquitous, Netflix and YouTube for entertainment, Yelp for restaurants, Amazon and Shein for fashion. Recommender system research has come a long way — accessing similarity using word matching to matching using embeddings. Content-based and collaborative filtering (CF) are the two primary schools of generating recommendations [3, 9]. Content-based recommender systems use the similarity between an item's attributes and a user's preference over them to recommend new items, for example recommending *Top Guns* to a user who likes *Action* movies [11, 4]. Collaborative filtering (CF) methods instead rely on the wisdom of the crowd to generate recommendations. Such methods are trained only using the ratings provided by the users of a recommendation platform like Netflix. The idea is that if two users have a similar rating pattern, then the items liked by a user can be recommended to the other (if they have not already interacted with it) [28, 32]. Hybrid recommender systems aim at combining the two approaches [10].

In recent years, CF has been widely chosen over content-based methods owing to 1) ease of training – one does not need to know the attributes of an item to incorporate it into the system, thereby eliminating the delay and reducing costs, and 2) better recommendations in terms of serendipity and satisfaction. However, this comes with one major limitation. Content-based methods are transparent and scrutable as they generate recommendations using a user's attribute preference, but this is not the case with CF-based methods. In such methods, users and items are mapped to an embedding space, which is learned from the user-item rating matrix (consisting of all users and all items) – and the

Submitted to 2022 Trustworthy and Socially Responsible Machine Learning (TSRML 2022). Do not distribute.

proximity in this space is used to generate recommendations. Such an embedding space is inherently uninterpretable. In its raw form, it is not feasible to understand what a model learned about the user's preference over the items and how it generates the recommendations.

Previous research has established that providing reasons for a recommendation enhances the transparency, scrutability, trustworthiness, effectiveness, persuasiveness, efficiency, and satisfaction of the recommender systems [35, 8, 31]. Providing explanations when using CF-based methods is nontrivial. This has spurred a lot of research in the broad field of "explainability for CF recommender systems". Most of the previous explanation generating approaches provide explanations in the form of either user-based or item-based explanations. User-based explanations explain a recommendation on the basis of 'similar' users liking it. And item-based explanations explain a recommendation on the basis of its closeness to other items that the user has liked in the past. Item-based explanations are usually easier to grasp as the user knows about the items they interacted with in the past. However, both of these explanation formats do not capture a user's preference over the attributes of an item — which is how users inherently think about a recommendation [18, 40, 16, 21, 22, 41]. (Providing attribute-based explanations is much easier when using content-based recommender systems, as users' preference is what those systems directly use to generate recommendations.)

This work proposes a novel approach that generates attribute-based explanations for CF-based recommender systems. A recommendation is explained in terms of a user's preference over the attributes of this item, e.g., 'We are recommending you this movie because you like *Action* movies.' Such explanations are personalized to the user and hence help further enhance the persuasiveness and trustworthiness of the recommender systems.

Previous research has also established the significance of post-hoc explainability, citing two reasons: 1) to evade the accuracy-interpretability trade-off (simpler and interpretable models are usually less accurate), and 2) generalizability to all recommender systems [41, 26, 23]. Aligning with this school of thought, our proposed approach is post-hoc and model-agnostic – it can generate explanations for any recommender system that operates using user and item embeddings.

We found this important intersection very under-researched in the literature. We could only find two methods that generate attribute-based explanations, LIME-RS [24] and AMCF [25]. LIME-RS did not evaluate its attribute-based explanations, and the metrics used by AMCF were not justified and not generalizable (see Appendix A). To this end, we also propose a set of generalizable and well-justified metrics to evaluate attribute-based explainability techniques for recommender systems (Section 4).

In summary, our contributions are:

1. We propose a novel technique to provide attribute-based explainability for recommender systems, a largely neglected research area.
2. Our technique is post-hoc and hence model-agnostic.
3. We also propose ten novel metrics to evaluate attribute-based explainability methods for recommender systems that can be used to compare all such future techniques.

2 Related Work

Machine Learning (ML) is being increasingly used to automate decisions. Some of the applications where ML is being used are highly critical and directly affect humans, for example, loan approval [30], criminal justice [34], and hiring [29]. The nascent field of trustworthy ML aims to detect bias in ML models (and counteract it), understand the factors that the ML model is using in making predictions, ensure the models respect privacy and security, and frame policies and regulations that the ML models should abide by [6, 5, 44]. In this work, we focus on explainability and refer the readers to other works for a broad discussion of trustworthy ML [6], [15], [38].

2.1 Explainability in Recommender Systems

Explainable recommender systems provide recommendations along with a justification for doing so. The term 'explainable recommendation' was introduced by Zhang et al. [48] in 2014, however there were papers talking about the benefits of providing explanations much earlier [36, 37, 31].

As mentioned, most previous methods for explainability in recommender systems provide explanations in user-based or item-based fashion [47]. The approaches that generate these explanations can either be model-specific or model agnostic, the former constituting the majority of previous methods. Models-specific approaches train inherently interpretable models by constraining the embedding space [26]. Zhang et al. [48] proposed extracting user preferences over attributes from item reviews and using that to provide explanations. They train an explicit factor recommender model that takes the user's preference over attributes as an input during training. Abdollahi and Nasraoui [1] add constraints in the latent space of the matrix factorization models to encourage explainability – they provide user and item-based explanations. There exist many other approaches in this category [13, 12, 7, 33, 19, 21]. A major downside of these approaches is that they are *not* post-hoc, i.e., they train a separate model for each recommendation use case, and this is problematic for two primary reasons: 1) there is a potential accuracy-interpretability trade-off, and 2) industrial teams have fine-tuned their recommender models over large datasets, and are rarely willing to train a new model from scratch in order to provide explanations. Hence post-hoc explainability techniques attract lot of attention, especially from industry perspective [39, 23].

2.2 Post-hoc Explainability in Recommender Systems

There only exist a handful of explainability approaches in this category. Peake and Wang [26] use a data-mining approach to provide item-based explanations in a post-hoc manner. Nóbrega and Marinho [24] propose LIME-RS which is motivated from the LIME paper [27]. It samples items close to a recommended item, learns a linear regression model on these samples, and uses the regressor to provide an explanation that can be either item-based or attribute-based (the latter can be provided if the attributes of the items are appended to the items embeddings when training the regression model). Cheng et al. [14] propose a technique that uses influence functions to find the influence of training rating on a particular recommendation, and the ratings with the highest influence are served as explanations. This technique, therefore, provides item-based explanations.

2.3 Attribute-Based Explainability in Recommender Systems

Unlike user- and item-based explanations, attribute-based explanations learn a user's personalized attribute preferences, thereby increasing the system's persuasiveness and trustworthiness [42, 16]. As mentioned earlier, there do exist works that provide explanations which utilize the user's preference over interpretable attributes of an item. Most of these works utilize the reviews provided by the users to capture this preference (and use that to train the recommender system itself [48, 21, 18]). However, such reviews might not exist even in the presence of ratings, for example, Netflix scrapped its review section due to low user participation recently [46]. Secondly, there is no guarantee of mention of interpretable attributes in the reviews that can be used to learn the user's preference. Even the datasets used in these papers had very sparse user reviews; over 77% of the users had only *one review* [17], making the inference of users' preferences questionable.

On the other hand, our approach also provides attribute-based explanations by utilizing the attributes of an item available in most recommender datasets (e.g., movie or game genres), and it *does not* use reviews. Similar to ours, AMCF [25] learns users' preferences over item attributes by training an attention network. So it provides attribute-based explanations like our approach; however, AMCF is not a post-hoc technique. Similarly, Wang et al. [42] proposes CERec to determine the attributes that are important for a user-item pair (in a counterfactual manner); however, CERec is also not a post-hoc technique. Wang et al. [41] propose a post-hoc RL-based approach that generates attribute-based explanations for a user; however, like most previous approaches, it requires access to the user reviews.

Hence our approach sits at the intersection of attribute-based (*without* using user reviews) and post-hoc explanation techniques. To the best of our knowledge, LIME-RS is the only previous approach that falls in this category.

3 Our Approach

Figure 1 shows the architecture of our approach. We train a model (*Auxiliary Model* in the diagram) that takes as input a user's embedding from the trained recommender model and an item's attribute vector. This attribute vector is multi-hot – it has 1s when the item has those attributes, otherwise, 0s

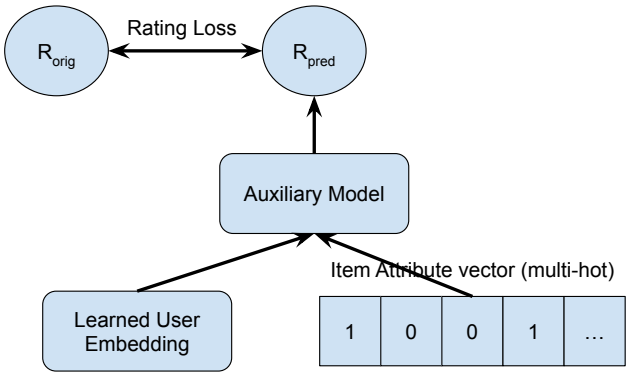

Figure 1: Architecture of our approach.

(instead of binary values, the vector could also have continuous values). The model is trained for all ratings present in the training dataset, and the goal is to reproduce the rating as given by the user (R_{orig} in the diagram). The model is trained using a mean-squared error between the original rating and the predicted rating (R_{pred} in the diagram). The auxiliary model can be instantiated with any ML model. For experiments, we used three models: Linear, a 2-layer neural network, and a 4-layer neural network. We train the model using Adam and SGD optimizers until convergence.

3.1 Generating Explanations

Once the auxiliary model is trained, we use it to generate explanations for recommended items. For a given user-item pair for which an explanation needs to be generated, our approach produces a sequence of that item's attributes ranked in decreasing order of that user's preference. We term this a user's *specific preference* over an item's attributes. The preference of a particular attribute (for a given user-item pair) is determined as the loss in predicted rating if that attribute was not there in the item. For example consider a movie who genres are *Crime*, *Documentary*, and *Horror*. For a user Alice, the auxiliary model predicts a rating of 4.2 for this movie. Now, we can use the auxiliary model to predict this movie's rating if the genres are zeroed out one by one. Table 1 shows this procedure. We zero out the attributes present in the item, one by one, and see the drop in rating (last column). The higher the drop, the higher the rank of the removed attribute. Our approach of attributing importance to attributes falls in a well-studied domain of explainability – broadly termed as removal-based explanation methods. We provide more details in Appendix A. In the example, the ranked order of preference is *Crime*, *Horror*, and *Documentary*. This is our analog of *local interpretability* in the context of classification in ML provided by popular methods like LIME [27] and SHAP [20].

Table 1: Illustration of how our approach computes the specific preference of a user over an item's attributes (in this case, genres of a movie).

Attribute zeroed	Predicted rating	Δ(Predicted rating)
No attribute zeroed	4.2	–
Horror	3.7	0.5
Documentary	3.9	0.3
Crime	3.2	1.0

We also generate the *general preference* of a user over all attributes in a dataset by accumulating the difference in predicted ratings when a particular attribute is zeroed over all the items that this user 'liked' in the past (note that this is not the same as all items that a user rated in the past). This is our analog of *global interpretability* in the context of classification in ML.

4 Evaluation

We performed experiments to answer the following research questions:

RQ1 How many of the test-set items that a user liked can the general preferences identified by an approach explain?

RQ2 How many of the top-k recommendations to a user can the general preferences identified by an approach explain?

RQ3 How user personalized are the specific and general preferences identified by an approach?

4.1 Experimental Methodology

Dataset. We used the popular recommender dataset Movielens-$100K$ for our experiments. Movielens-$100K$ has 100,000 ratings for 1682 movies provided by 943 users. Each rating is an integer from 1 to 5, with a higher number meaning a higher likeness of it. The dataset also contains the genre of each movie which is a multi-hot vector of size 18: there are a total of 18 genres in the dataset: *Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western* and each movie is categorized as having one or more genre. These are the attributes that a technique identifies a user's preference over. We used 70% of the dataset for training a matrix factorization-based recommender system and 30% of the data as the test set (data split is stratified by users).

Metrics. As mentioned in Section 2, there are only two previous methods that can generate attribute-based explanations for recommender systems: LIME-RS [24] and AMCF [25]. However, LIME-RS did not quantitatively evaluate its attribute-based explanations, and the AMCF did not justify its metric of choice (see Appendix A). Hence, we propose a set of 10 generalized metrics to evaluate any attribute-based explanation method for recommender systems. We evaluate both general and specific preferences in these metrics.

1. *Test set coverage:* This metric finds if the general preferences of a user identified by a technique have any intersection with the attributes of the items in the test set that they liked. For example, suppose the identified general preferences for a user Adam are *Anime*, *Comedy*, and *Sci-Fi* and there is a movie that Adam liked whose genres are *Action* and *Anime*. In that case, we could count this movie as covered. Had the genre of the movie only been *Action*, then it would not have been covered. For Movielens-$100K$, we consider a movie 'liked' if a user has rated it 4 or 5. We only consider the top-3 general preferences for a user when measuring coverage. We report the mean coverage over all the users for this metric.
2. *Top-k recommendations coverage:* This metric finds if the top-3 identified general preferences covers (i.e., have any intersection) with the attributes of the top-20 recommended items to a user. We report the mean coverage over all the users for this metric.
3. *Personalization of the explanations:* Since a few attributes in most recommender datasets are very popular, i.e., they occur in almost all items: identifying such an attribute as a user's preference will provide almost 100% coverage for the metrics mentioned above – however, this might be an inaccurate and unpersonalized explanation. To measure how personalized the explanations are, we need to know the ground-truth users' preferences over the set of attributes – unfortunately, this is unknown to us. To overcome this limitation, we propose two methods that act as reasonable proxies for a user's attribute preference (The gold standard would be to conduct a user study; however, that would involve collecting a new recommender dataset and asking users to tell us about the preferences, which we leave to future work.) :
   (a) *Conditional Probability of Liking given a genre is present:* This measures the probability that a user likes a movie, given that the attribute is present in a movie. It is the ratio of the number of times a genre is present in the movies a user likes (rated 4 or 5) and the number of times it is present in all the movies the user rated (rated 1 through 5).
   (b) *Odds of Liking vs. Disliking:* This measures the ratio of the number of times a genre is present in a movie a user likes (rated 4 or 5) and the number of times it is present in a movie that the user dislikes (rated 1 or 2).
   We use the training set for calculating both these preference proxies.

   Both provide a reasonable proxy of users' general preference over genres, i.e., weights over the set of genres. We report four metrics considering such weights as ground truth user preference:

   (a) *General preferences coverage:* This metric measures if there is any intersection between the top-k general preferences identified for a user and the top-k preferences identified by either of the two proxies mentioned above.

(b) *General preferences ranking:* This metric measures the similarity between the ranking of the top-k general genre preferences identified for a user and the top-k preferences identified by either of the two proxies. We use rank-biased overlap (RBO) as a measure of similarity between the two ranked lists (we justify this choice in Appendix A).

(c) *Specific preferences coverage:* Similar to general preferences, we also measure if there is any intersection between the top-k genre preferences for each item that a user liked in the training set (specific genre preference for a user-item pair) and the top-k genre preferences identified for this user by either of the two proxies.

(d) *Specific preferences ranking:* We also measure the similarity in the ranking of the top-k genre preferences for items liked by a user in the training set and the top-k genre preferences identified for this user by either of the two proxies.

For all the above metrics, we report the mean over all the users. Hence we have a total of 8 metrics for measuring the personalization of the explanations. We used k = 3 in the experiments.

Baselines. We compare our approach to 6 baseline methods:

- *LIME-RS [24]:* The only previous post-hoc attribute-based explainability method.
- *AMCF [25]:* Another attribute-based explainability method – this is not a post-hoc method.
- *AMCF-PH:* The AMCF approach adapted for being post-hoc (details in Appendix A).
- *Global popularity:* The genres that are the most popular among all the movies that are liked across the entire dataset. For Movielens-$100K$, these were *Action*, *Comedy*, and *Drama*.
- *User-specific popularity:* The most popular genres for a particular user. These are calculated based on only the movies they liked (rated 4 or higher).
- *Random:* This baseline is for control, for each metric, it selects a random set of genres for each user-item pair, when computing specific preferences, and for each user when computing the general preferences.

Table 2: The test set coverage, top-k recommendations coverage, and the 8 explanation personalization metrics are reported here (averaged over all users). For our method, we computed metrics with three auxiliary models: Linear, a 2-layer neural network (MLP1), and a 4-layer neural network (MLP2). For all columns, a higher value is better. The highest value for each metric is emboldened.

Metrics	LIME-RS	AMCF	AMCF-PH	GBL Popl.	User Popl.	Random	Our-Linear	Our-MLP1	Our-MLP2
Testset Coverage	8.4	47.2	52.6	**84.5**	77.9	32.4	60.7	70.8	71.2
Recommendations Coverage	26.0	44.3	46.1	**73.2**	69.6	30.6	61.3	68.6	67.8
CondProb Generalpref Coverage	50.3	58.2	47.2	29.0	41.8	43.9	**62.9**	58.1	56.6
CondProb Generalpref Ranking	**15.8**	15.1	11.2	5.8	9.9	11.0	15.2	14.0	13.8
CondProb Specificpref Coverage	52.5	53.1	52.4	29.0	41.7	44.1	**55.3**	53.8	53.6
CondProb Specificpref Ranking	14.6	49.6	50.3	6.8	10.4	11.0	**52.2**	50.7	51.2
Odds Generalpref Coverage	32.6	63.7	60.4	48.5	55.0	43.7	**73.4**	72.3	70.4
Odds Generalpref Ranking	9.4	19.4	18.0	18.2	26.7	10.6	25.6	27.6	**28.2**
Odds Specificpref Coverage	37.2	55.8	56.8	48.5	55.0	44.2	**59.8**	59.2	59.5
Odds Specificpref Ranking	10.2	49.7	50.7	19.6	27.2	11.0	50.8	51.6	**51.7**

4.2 Answers to our research questions:

Table 2 reports all the metrics for all the baselines and our approach. We use it to answer our research questions:

1. *LIME-RS:* The test set and top-k recommendations coverage provided by LIME-RS is abysmally low, even worse than the *random* baseline. It trails our approach in all the 10 metrics, barring the ranking of general preferences when the ground truth preference is calculated using the conditional probability of liking.
2. *AMCF:* This approach trails our approach in all metrics.
3. *AMCF-PH:* The post-hoc adapted version of AMCF also trails our approach in all metrics.
4. *Global popularity:* It achieves the highest test set and top-k recommendations coverage among all techniques, and understandably so — the three most common genres *Action*, *Comedy*, and *Drama* occur is over 88% of the test set items (see the feature distribution plot in Figure 3). However, it

performs much worse than our approach on all personalization metrics, many times even worse than the *random* baseline. Therefore, we can conclude that the explanations served by this approach do not capture a user's preference over genres and can perform well for coverage-based metrics because of their skewed distribution.

5. *User-specific popularity:* This popularity approach achieves the second highest test set and top-k recommendations coverage, and similar to *global popularity* performs worse than ours on all personalization metrics, many times even worse than the *random* baseline.
6. *Random:* We used this approach to serve as control and to ensure that no metric was trivial to perform well on. Not surprisingly, it performs worse than our approach on all metrics.
7. *Ours:* Our approach has the third highest test set and top-k recommendations coverage (approaching the performance of user-specific popularity for the latter) while performing the best on all personalization metrics except for the general preferences ranking when using the conditional probability of liking, where it performs the second best. Therefore, we can conclude that our approach serves the most personalized explanations while still being able to explain a large proportion of test set items and top-k recommendations.

4.3 Discussion:

As our metrics indicate, our approach strikes a great balance between coverage and personalization of the explanations. We would also like to mention that neither of the previous techniques, LIME-RS or AMCF, considered popularity as a baseline. We show that popularity approaches perform pretty well on the two coverage metrics (those are the metrics that were used in their papers). Even the item-based explainability papers [1, 26] did not compare to popularity even though their metrics were precision and recall. Since recommender systems are known to have popularity bias [2, 49], popular items can perform very well on these metrics. Hence we consider discussing popularity as a potential explanation for both item-based and attribute-based explanations as a contribution of our work.

References

[1] Behnoush Abdollahi and Olfa Nasraoui. 2017. Using Explainability for Constrained Matrix Factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 79–83. https://doi.org/10.1145/3109859.3109913

[2] Himan Abdollahpouri, R. Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-ranking. *ArXiv* abs/1901.07555 (2019).

[3] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (jun 2005), 734–749.

[4] Charu C Aggarwal. 2016. Content-based recommender systems. In *Recommender systems*. Springer, 139–166.

[5] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. https://doi.org/10.48550/ARXIV.1606.06565

[6] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. *Fairness and Machine Learning*. fairmlbook.org. http://www.fairmlbook.org.

[7] Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 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 (KDD '17)*. Association for Computing Machinery, New York, NY, USA, 9. https://doi.org/10.1145/3097983.3098170

[8] Mustafa Bilgic and Raymond Mooney. 2005. Explaining Recommendations: Satisfaction vs. Promotion. In *Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces*. http://www.cs.iit.edu/~ml/pdfs/bilgic-iui05-wkshp.pdf

[9] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems* 46 (2013), 109–132.

[10] R. Burke. 2004. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12 (2004), 331–370.

[11] Laurent Candillier, Kris Jack, Françoise Fessant, and Frank Meyer. 2009. State-of-the-art recommender systems. In *Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling*. IGI Global, 1–22.

[12] 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 and Development in Information Retrieval (SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 4. https://doi.org/10.1145/3209978.3210083

[13] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to Rank Features for Recommendation over Multiple Categories. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/2911451.2911549

[14] 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 and Data Mining (KDD '19)*. Association for Computing Machinery, New York, NY, USA, 9. https://doi.org/10.1145/3292500.3330857

[15] Ian Covert, Scott Lundberg, and Su-In Lee. 2021. Explaining by Removing: A Unified Framework for Model Explanation. *Journal of Machine Learning Research* 22, 209 (2021), 1–90. http://jmlr.org/papers/v22/20-1316.html

[16] Bruce Ferwerda, Kevin Swelsen, and Emily Yang. 2018. Explaining Content-Based Recommendations. (2018), 26. https://bruceferwerda.com/docs/Ferwerda_DigitalTVGuide.pdf

[17] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. TriRank: Review-Aware Explainable Recommendation by Modeling Aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*. Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/2806416.2806504

[18] Yunfeng Hou, Ning Yang, Yi Wu, and Philip S. Yu. 2019. Explainable Recommendation with Fusion of Aspect Information. *World Wide Web* 22, 1 (2019), 20. https://doi.org/10.1007/s11280-018-0558-1

[19] Huafeng Liu, Jingxuan Wen, Liping Jing, Jian Yu, Xiangliang Zhang, and Min Zhang. 2019. In2Rec: Influence-Based Interpretable Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/3357384.3358017

[20] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[21] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. Association for Computing Machinery, New York, NY, USA, 8. https://doi.org/10.1145/2507157.2507163

[22] Julian McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. *2012 IEEE 12th International Conference on Data Mining* (2012), 1020–1025.

[23] Christoph Molnar. 2022. *Interpretable Machine Learning* (2 ed.). https://christophm.github.io/interpretable-ml-book

[24] Caio Nóbrega and Leandro Marinho. 2019. Towards Explaining Recommendations through Local Surrogate Models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19)*. Association for Computing Machinery, New York, NY, USA, 8. https://doi.org/10.1145/3297280.3297443

[25] Deng Pan, Xiangrui Li, Xin Li, and Dongxiao Zhu. 2020. Explainable Recommendation via Interpretable Feature Mapping and Evaluation of Explainability. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20 (IJCAI'20)*. International Joint Conferences on Artificial Intelligence Organization, 2690–2696. https://doi.org/10.24963/ijcai.2020/373 Main track.

[26] 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 and Data Mining (KDD '18)*. Association for Computing Machinery, New York, NY, USA, 2060–2069. https://doi.org/10.1145/3219819.3220072

[27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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 (KDD '16)*. Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/2939672.2939778

[28] Abhijit Roy. 2020. Introduction To Recommender Systems- 1: Content-Based Filtering And Collaborative Filtering. https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421. [Online; accessed 28-December-2021].

[29] Kumba Sennaar. 2019. Machine Learning for Recruiting and Hiring – 6 Current Applications. https://emerj.com/ai-sector-overviews/machine-learning-for-recruiting-and-hiring/. Accessed: 2020-10-15.

[30] Saurav Singla. 2020. Machine Learning to Predict Credit Risk in Lending Industry. https://www.aitimejournal.com/@saurav.singla/machine-learning-to-predict-credit-risk-in-lending-industry. Accessed: 2020-10-15.

[31] Rashmi Sinha and Kirsten Swearingen. 2002. The Role of Transparency in Recommender Systems. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems (CHI EA '02)*. Association for Computing Machinery, New York, NY, USA, 830–831. https://doi.org/10.1145/506443.506619

[32] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).

[33] Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. 2019. The FacT: Taming Latent Factor Models for Explainability with Factorization Trees. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/3331184.3331244

[34] Jason Tashea. 2017. Courts Are Using AI to Sentence Criminals. That Must Stop Now. https://www.wired.com/2017/04/courts-using-ai-sentence-criminals-must-stop-now/. Accessed: 2020-10-15.

[35] Nava Tintarev. 2007. Explanations of Recommendations. In *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys '07)*. Association for Computing Machinery, New York, NY, USA, 203–206. https://doi.org/10.1145/1297231.1297275

[36] Nava Tintarev and Judith Masthoff. 2007. A Survey of Explanations in Recommender Systems. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*. 801–810. https://doi.org/10.1109/ICDEW.2007.4401070

[37] Nava Tintarev and Judith Masthoff. 2011. *Designing and Evaluating Explanations for Recommender Systems*. Springer US, Boston, MA, 479–510. https://doi.org/10.1007/978-0-387-85820-3_15

[38] Kush R. Varshney. 2021. *Trustworthy Machine Learning*. Chappaqua, NY, USA. http://trustworthymachinelearning.com.

[39] Sahil Verma, Keegan Hines, and John P Dickerson. 2021. Amortized Generation of Sequential Counterfactual Explanations for Black-box Models. , arXiv–2106 pages.

[40] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: Explaining Recommendations Using Tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI '09)*. Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/1502650.1502661

[41] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018. A Reinforcement Learning Framework for Explainable Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. 587–596. https://doi.org/10.1109/ICDM.2018.00074

[42] Xiangmeng Wang, Qian Li, Dianer Yu, and Guandong Xu. 2022. Reinforced Path Reasoning for Counterfactual Explainable Recommendation. https://doi.org/10.48550/ARXIV.2207.06674

[43] William Webber, Alistair Moffat, and Justin Zobel. 2010. A Similarity Measure for Indefinite Rankings. *ACM Trans. Inf. Syst.* 28, 4, Article 20 (nov 2010), 38 pages. https://doi.org/10.1145/1852102.1852106

[44] Adrian Weller. 2017. Transparency: Motivations and Challenges. https://doi.org/10.48550/ARXIV.1708.01870

[45] Wikipedia contributors. 2022. Rank correlation — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Rank_correlation&oldid=1087086457 [Online; accessed 25-September-2022].

[46] Emma Woollacott. 2018. Why Netflix Won't Let You Write Reviews Any More. https://www.forbes.com/sites/emmawoollacott/2018/07/06/why-netflix-wont-let-you-write-reviews-any-more/?sh=63af0ffe6570 [Online; accessed 25-September-2022].

[47] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Found. Trends Inf. Retr.* 14 (2020), 1–101.

[48] 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 and Development in Information Retrieval (SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/2600428.2609579

[49] Ziwei Zhu, Yun He, Xing Zhao, and James Caverlee. 2021. Popularity Bias in Dynamic Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*. Association for Computing Machinery, New York, NY, USA, 11. https://doi.org/10.1145/3447548.3467376

A Appendix

AMCF Evaluation Metrics: AMCF also evaluate their general and specific preference. For general preference, similar to our evaluation metrics, their primary metric is *Top M recall at K*, i.e., the intersection between the top M ground truth user preference and top K predicted user preferences. To realize this metric, they need the ground truth user preference – which is not present in the dataset – and hence similar to our technique, they need to simulate. The way they compute the ground truth preference is unjustified and inexplicable. The process involves computing the weight of each item

by removing the user and item bias terms, and then the preference of an attribute is just the sum of the weights of the items it occurs in. The latter part is still reasonable; however, we are uncertain about the weight calculation part. For evaluating specific preferences, they use the sorted order of general preference for the attributes present in that specific item – which does not resolve the concerns mentioned above.

We also use proxies to simulate the users' ground truth attribute preference; however there are key differences in our evaluation when compared to AMCF:

- The proxies are more generalizable and reasonable, like the conditional probability of liking and odds of liking.
- We use multiple proxies and report results on all of them to avoid cherry-picking.

Justification of the removal-based explanation: Covert et al. [15] developed a framework to categorize such methods along three dimensions:

- Attribute removal: how the approach removes attributes from the model,
- Model behavior: what model behavior is it observing, and
- Summary technique: how does it summarize an attribute's impact?

Our approach, when instantiated in this framework: Removes attributes by setting them to zero Analyzes prediction as the model behavior Summarizes an attribute's impact by removing them individually Since our attribute input vector is binary and indicates whether an attribute is present or not, simulating an attribute's removal by setting it to zero is a natural choice. Previous removal-based explanation methods have used *prediction* or *prediction loss* or *dataset loss* for model behavior analysis. We chose prediction for analysis because we wanted to get specific preference (analog of local interpretability) without needing the correct rating. Previous removal-based explanation methods have used *removing individual attributes* or *Shapley values* or *trained additive models* to get attribute impact value. Removing individual attributes accesses the impact of an attribute by measuring the loss in prediction when that one attribute is removed, and this is what we choose. On the other hand, Shapley values take all subsets of attributes and then use the cooperative game theoretic formulation to assign impact value to each attribute. It has two disadvantages:

- It creates all subsets of attributes – which is exponential in the number of attributes, making the process very expensive.
- For creating all the subsets, it simulates removing many features that can potentially create attribute vectors that the auxiliary model has not seen, and thereby its prediction can not be trusted in that part of the data manifold.

For these reasons, we choose the method of removing individual attributes. There are potential downsides to this choice as well. If the ground truth reason for recommending a movie is either if it were either *Horror* or *Crime*, zero-ing out one of them will not make a difference in prediction, and hence their assigned impact will not be correct.

Justification of using Rank-Biased Overlap (RBO):

Comparing ranked lists can be achieved using various rank correlation metrics like Kendall's Tau and Spearman's correlation [45]. These metrics have restrictions that the ranked lists must be conjoint, i.e., they both must contain all the items that the entire universe of items contains. This is not suitable for our metrics because the top-k attributes from either the conditional probability or the odds liking proxy might not have any common element with the ranking produced by any technique. Hence we choose to compare the ranked lists using rank-biased overlap (RBO) metric [43], which was recently proposed to overcome the limitation posed by correlation-based metrics. Specifically, RBO is better than correlation-based metrics as:

- RBO does not require the two ranked lists to be conjoint, i.e., a ranked list having items that do not occur in the other list is acceptable, e.g., the similarity between [1, 3, 7] and [1, 5, 8] can be measured using RBO.
- RBO does not require the two ranked lists to be of the same length.
- RBO provides weighted comparison, i.e., discordance at higher ranks is penalized more than discordance at lower ranks.

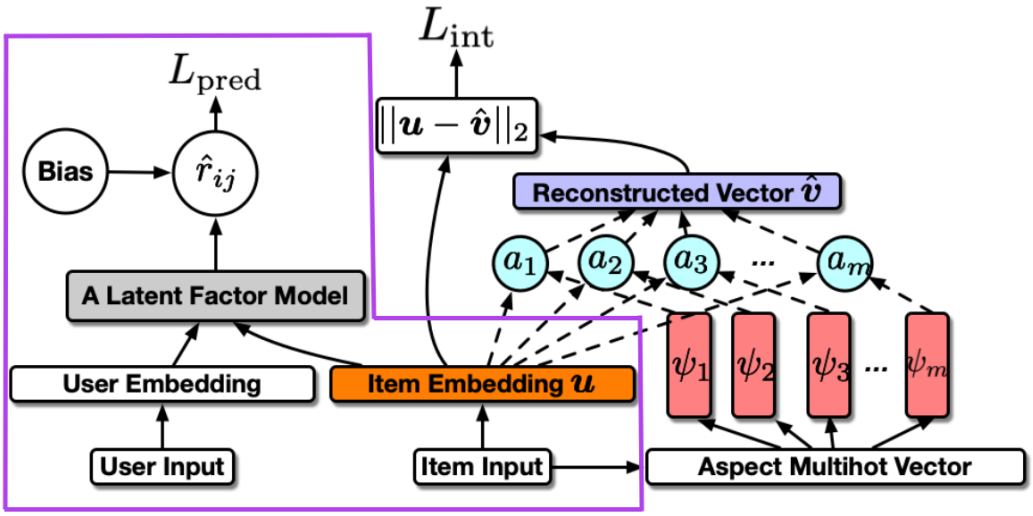

Figure 2: The architecture of AMCF Post-Hoc (AMCF-PH). The part within the blue colored region was not trained, thereby making the technique post-hoc.

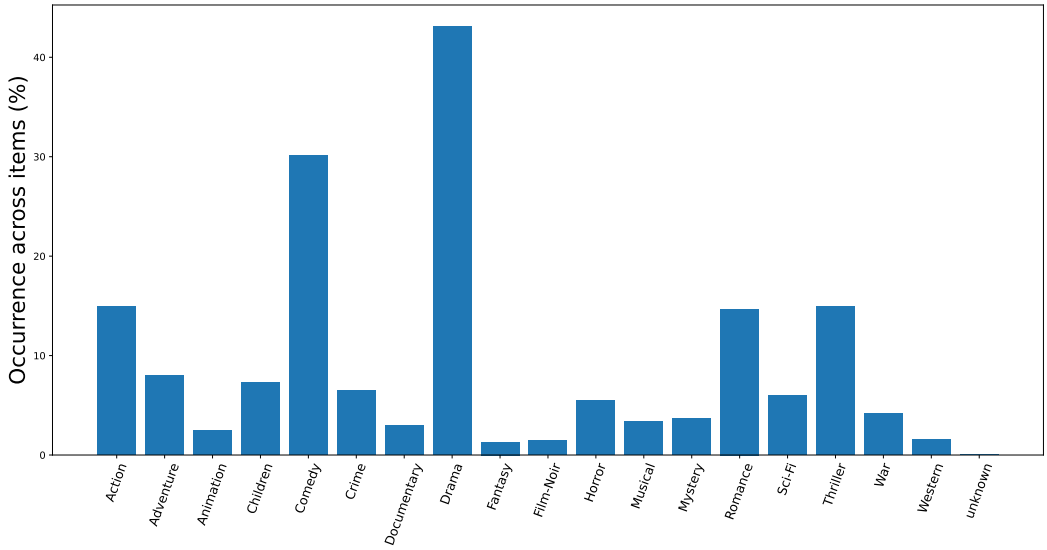

Figure 3: The genre distribution for Movielens-100K dataset. The top-3 most popular genres: *Action*, *Comedy*, and *Drama* occur in over 88% of the movies.

AMCF Post-Hoc Adaptation:

For adapting AMCF [25] to be post-hoc, we made a minor modification to the architecture mentioned in their paper (see Figure 2). We froze the left-hand side of the architecture and only trained the right-hand side, which consists of the attention network that aims to reconstruct the item's embedding from its attributes. Since the user and item embeddings were not trained, this made the technique post-hoc.

Figure 3 shows the genre distribution for Movielens-100K.