Enhancing Explainability and Scrutability of Recommender Systems

Azin Ghazimatin[1]

Abstract: Our increasing reliance on complex algorithms for recommendations calls for models and methods for explainable, scrutable, and trustworthy AI. While explainability is required for understanding the relationships between model inputs and outputs, a scrutable system allows us to modify its behavior as desired. These properties help bridge the gap between our expectations as end users and the algorithm's behavior and accordingly boost our trust in AI. Aiming to cope with information overload, recommender systems play a crucial role in filtering content (such as products, news, songs, and movies) and shaping a personalized experience for their users. Consequently, there has been a growing demand from the information consumers to receive proper explanations for their personalized recommendations. To this end, we put forward proposals for explaining recommendations to the end users. These explanations aim at helping users understand why certain items are recommended to them and how their previous inputs to the system relate to the generation of such recommendations. Such explanations usually contain valuable clues as to how a system perceives user preferences and more importantly how its behavior can be modified. Therefore, as a natural next step, we develop a framework for leveraging user feedback on explanations to improve their future recommendations. We evaluate all the proposed models and methods with real user studies and demonstrate their benefits at achieving explainability and scrutability in recommender systems.

Keywords: Recommender Systems; Explainable AI; Scrutability

1 Introduction

Our increasing reliance on complex algorithms for recommendations calls for models and methods for explainable and scrutable AI. While explainability helps us understand the cause of a decision made by an algorithm [Mi19], a scrutable system enables users to correct system's assumptions when needed [TM07]. These properties bring about trust by bridging the gap between humans and AI.

Aiming to personalize content based on user preferences, recommender systems are perceived as advice-givers that can improve our acceptance through explanations [RRS15]. With the emergence of more complex models [KBV09] outperforming the simpler and more explainable ones [Sa01], *Explainable AI* has progressively received more attention from the Recommender Systems (RecSys) community [ZC20]. Lack of transparency in recommender systems can have a direct impact on user acceptance, as based on the content personalized

[1] Saarland University and Max Planck Institute for Informatics, 66123 Saarbrücken, Germany aghazima@mpi-inf.mpg.de

doi:10.18420/BTW2023-32

for users, they may feel that the system is labeling them inappropriately[2] or misusing their private information[3]. To highlight the gravity of this matter, recently, laws have been passed to establish users' right to explanations [GF17].

Despite the close tie between explainability and scrutability [BR20], they do not necessarily entail each other. In other words, knowing why the algorithm makes particular choices may not be sufficient for realizing how to modify it. For instance, imagine a user of an online movie streaming service who is frequently recommended with action movies. The system explains its choices by drawing connections between the recommended movies and the action movies the user previously watched on the platform. Now, consider a situation where the user wants to stop receiving such movies as they do not entirely match her interest. Here, the provided explanations do not act as a precise guide as to how they can effectively exert control over their recommendations. Therefore, scrutability in recommender systems requires separate consideration and handling.

The following sections delve into the concepts of explainability and scrutability and describe our contributions towards realizing these objectives.

2 Explainable Recommendations

Recommender systems aim at delivering personalized content such as products, movies, books, and songs to their users. The chosen content is often visualized in a ranked list, where the order reflects the relevance of the items to the user. To compute these relevance scores, recommender systems usually train models based on various inputs collected from their users. User inputs can be explicit (e.g., rating or liking an item) or implicit (e.g., watching a movie or listening to a song). The abundance of implicit signals has facilitated data collection by service providers.

Providing the systems with an enormous amount of data over time, users might not be able to remember all the details of their interactions, and hence experience difficulty in understanding why they receive certain items as their recommendations. This problem particularly worsens when users do not even have access to the complete history of their interaction with the system, a phenomenon referred to as *inverse privacy* [GW16]. Therefore, it is imperative for the recommender systems to be explainable, i.e., to enable users to understand the relationships between their own input to the system and the recommendations they receive.

To illustrate how a recommendation can be explained, imagine a user who is a member of a social cataloging website like Goodreads[4] and receives a book recommendation, titled *Recovery: Freedom from Our Addictions*. Example 2.1 presents a possible way of

[2] https://www.wsj.com/articles/SB1038261936872356908

[3] https://www.wired.co.uk/article/tiktok-filter-bubbles

[4] https://www.goodreads.com

explaining this recommendation to the user by outlining a connection between the given recommendation and their past actions on the platform:

Example 2.1 *You* $\xrightarrow{\textit{liked}}$ *Becoming* $\xrightarrow{\textit{has genre}}$ *Autobiography* $\xrightarrow{\textit{belongs to}}$ *Recovery: Freedom from Our Addictions*

In Section 2.1, we describe a framework for generating such explanatory paths based on user's interactions with a given platform.

Apart from describing *why* a certain item is relevant to a user, recommender systems are also expected to be able to explain the rankings, i.e., to reason *why* a certain item is *more relevant* than the others. For instance, the following statement explains the cause of receiving the book *Recovery: Freedom from Our Addictions* as the *top-ranked* recommendation:

Example 2.2 *You are recommended with the book Recovery: Freedom from Our Addictions because you liked the books* ***Becoming*** *and* ***Dreams from My Father****. If you did not like these two books, your top-ranked recommendation would be the book* ***Food and Nutrition****.*

Example 2.2 shows that *liking* the books *Becoming* and *Dreams from My Father* is the key reason that the book *Recovery: Freedom from Our Addictions* is more relevant to the user than the book *Food and Nutrition*. The blue text in this example demonstrates the causality between user's previous action and system's outcome. Such explanations are referred to as counterfactual; they pinpoint those user actions whose absence would result in a different recommendation for them. Identifying the true reasons behind the recommendations, these explanations pave the way towards scrutability, i.e., they help shed light on how users can control what they see as their recommendations. In Section 2.2, we describe a method for generating counterfactual explanations.

2.1 Post-hoc Explanations for Black-Box Recommenders

Web users interact with a huge volume of content every day, be it for news, entertainment, or inside social conversations. To save time and effort, users are progressively depending on curated *feeds* for such content. A feed is a stream of *individualized* content items that a service provider tailors to a user. One example of a feed is the list of questions and answers recommended to users on Quora[5]. Since a feed is a one-stop source for information, it is important that users understand *how items in their feed relate to their profile and activity on the platform*.

To help users understand these relationships, we introduce FAIRY, a **F**ramework for **A**ctivity-**I**tem **R**elationship discover**Y**. FAIRY enables users to discover useful and surprising

[5] https://www.quora.com

relationships between their own actions on the platform and their recommendations. For this, we first model the user's local neighborhood on the platform as an interaction graph. This graph is constructed solely from the information available to the user. In a user's interaction graph, the set of simple paths connecting the user to her feed item are treated as pertinent explanations. Example 2.1 illustrates one such explanatory path. Next, FAIRY scores the discovered explanations with learning-to-rank models built upon users' judgements on relevance and surprisal of the explanation paths. Longitudinal user studies on two social platforms, *Quora* and *Last.fm*[6], demonstrate the practical viability and user benefits of this framework in different domains. For more detailed analysis, refer to [Gh21a; GSW19].

2.2 Counterfactual Explanations for Recommendations

FAIRY's explanations are post-hoc, i.e., they are decoupled from the underlying recommender system. While essential for enhancing transparency of black-box models, these explanations are not actionable; they may mislead the user when used for modifying the system's behavior. To overcome this limitation, we introduce PRINCE, a method for **Pr**ovider-side **In**terpretability with **C**ounterfactual **E**vidence.

PRINCE enables graph-based recommenders with personalized PageRank at their core [NK19] to generate concise and counterfactual explanations for their users. To see an example of such explanations, see Example 2.2. PRINCE explains the most relevant recommendation to the user by identifying the minimum number of their previous actions whose removal from the user history could displace the top-ranked item. To find the minimal counterfactual explanations from an exponential search space, Prince uses a polynomial-time algorithm, and hence it is efficient.

Experiments on two real-world datasets show that PRINCE provides more compact explanations than intuitive baselines. Insights from a crowdsourced user-study demonstrate the viability of such action-based explanations. For further details refer to [**ghaz**; Gh20].

3 Scrutable recommendations

A *scrutable* recommender system allows its users to tell the system when it is wrong and enables users to steer their recommendations accordingly [TM07]. This feature is particularly useful when users experience drifts in their interests or when the system cannot correctly infer their preferences. Evidence suggests that scrutability can improve user's engagement level and their satisfaction [HKN12; Kn12; PB15].

Critique-enabled recommenders have already taken the first step towards scrutability. These systems employ a feedback mechanism called *critiquing* that enable users to express their

[6] `https://www.last.fm`

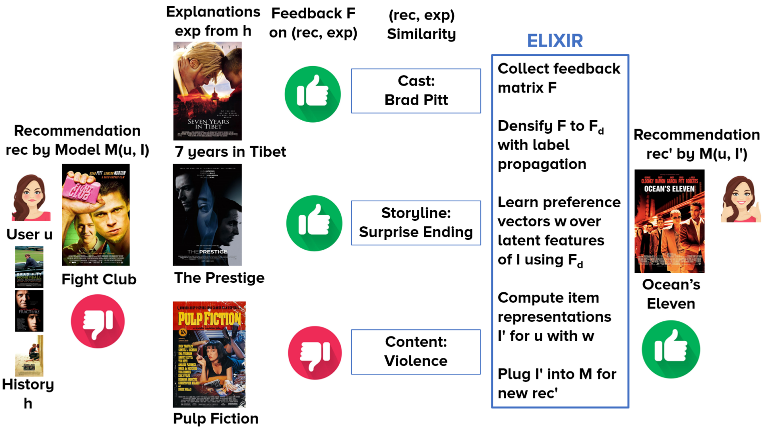
Explanations exp from h
Feedback F on (rec, exp)
(rec, exp) Similarity
ELIXIR
Recommendation rec by Model M(u, I)
User u
Fight Club
History h
7 years in Tibet
The Prestige
Pulp Fiction
Cast: Brad Pitt
Storyline: Surprise Ending
Content: Violence
Collect feedback matrix F
Densify F to F$_d$ with label propagation
Learn preference vectors w over latent features of I using F$_d$
Compute item representations I' for u with w
Plug I' into M for new rec'
Recommendation rec' by M(u, I')
Ocean's Eleven

Figure 1: Example illustrating the intuitions behind ELIXIR.

dissatisfaction with some characteristics of the recommended item [CP12]. For instance, imagine a student who relies on an online service like Yelp [7] to find a nice place to have dinner. The recommended restaurants, however, are not suitable for her as they are mostly expensive and far from her place. In this scenario, she will benefit from system-suggested critiques such as *show me a cheaper* or *closer restaurant* that enables her to explore other options that suit her interest better. In the next section, we describe how recommender systems can leverage user feedback on explanations as a critiquing mechanism to improve their future recommendations.

3.1 Using Explanations to Improve Recommender Systems

Explanations contain valuable information on *why* a certain item is recommended to the user. We posit that the similarity between the recommended item and is corresponding explanation speaks to the reason behind receiving the recommendation in the first place. This drives the design of a feedback collection mechanism to learn users' fine-grained preferences. Fig. 1 shows an illustrative scenario. User u receives a recommendation for the movie *Fight Club* (rec) based on her online history and factors like item-item similarities. This is accompanied by an explanation referring to three items, all previously liked by u and being similar, in some aspects, to rec. We have exp_1: *Seven Years in Tibet*, exp_2: *The Prestige*, and exp_3: *Pulp Fiction*. The system generated these three items for explanation because:

- exp_1 features the actor Brad Pitt who also stars in rec,
- exp_2 has a surprise ending, similar to rec,

[7] https://www.yelp.com

- exp_3 contains violent content, like rec.

Now suppose that user u loves Brad Pitt and surprise endings but hates disturbing violence (she likes *Pulp Fiction* for other reasons like its star cast and dark comedy, that dominated her opinion, despite the violence). When receiving rec with the above explanation, user u could give different kinds of feedback. The established way is to simply dislike rec, as a signal from which future recommendations can learn. However, this would completely miss the opportunity of learning from how user u views the three explanation items. The best feedback would be if user u could inform the system that she likes Brad Pitt and surprise endings but dislikes violence, for example, by checking item properties or filling in a form or questionnaire. However, this would be a tedious effort that few users would engage in. Also, the system would have to come up with a very fine-grained feature space of properties, way beyond the usual categories of, say, movie genres.

To facilitate efficient critiquing, we introduce ELIXIR, a framework for (**E**fficient **L**earning from **I**tem-based e**X**planations **I**n **R**ecommenders). ELIXIR enables recommenders to obtain user feedback on pairs of recommendation and explanation items, where users are asked to give a binary rating on the shared aspects of the items in a pair. To incorporate the collected feedback, we propose a method to learn user-specific latent preference vectors used for updating item-item similarities. The underlying intuition is to increase (decrease) the distance of disliked (liked) items and the like to the user's profile, such that the quality of future recommendations is improved. Our framework is instantiated using generalized graph recommendation based on personalized PageRank. Insightful experiments with a real user study show significant improvements for movie and book recommendations over item-level feedback. For a detailed analysis, refer to [Gh21a; Gh21b].

4 Conclusion

In this work, we studied explainability and scrutability of recommender systems. We introduced FAIRY, a framework for generating post-hoc explanations for black-box recommenders. We further proposed PRINCE, a provider-side interpretability tool, to provide users with concise and counterfactual explanations. Putting explanations into action, we lastly introduced ELIXIR, a framework for leveraging user feedback on explanations to improve their future recommendations. Our studies demonstrate the benefits of explanations for both end users and service-providers: users gain insight into the personalization process, and service providers enhance their users' experiences by offering more transparency and facilitating user control through feedback on explanations. We hope that this work sparks interest in the community towards responsible systems and pushes forward the mindsets and infrastructures required for trustworthy AI.

Literatur

[BR20] Balog, K.; Radlinski, F.: Measuring Recommendation Explanation Quality: The Conflicting Goals of Explanations. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020. S. 329–338, 2020, URL: https://doi.org/10.1145/3397271.3401032.

[CP12] Chen, L.; Pu, P.: Critiquing-based recommenders: survey and emerging trends. User Modeling and User-Adapted Interaction 22/1, S. 125–150, 2012.

[GF17] Goodman, B.; Flaxman, S. R.: European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation". AI Mag. 38/3, S. 50–57, 2017, URL: https://doi.org/10.1609/aimag.v38i3.2741.

[Gh20] Ghazimatin, A.; Balalau, O.; Saha Roy, R.; Weikum, G.: PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. In: WSDM ’20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020. S. 196–204, 2020, URL: https://doi.org/10.1145/3336191.3371824.

[Gh21a] Ghazimatin, A.: Enhancing explainability and scrutability of recommender systems, Diss., Saarland University, Saarbrücken, Germany, 2021, URL: https://publikationen.sulb.uni-saarland.de/handle/20.500.11880/32590.

[Gh21b] Ghazimatin, A.; Pramanik, S.; Roy, R. S.; Weikum, G.: ELIXIR: Learning from User Feedback on Explanations to Improve Recommender Models. In: WWW ’21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. S. 3850–3860, 2021, URL: https://doi.org/10.1145/3442381.3449848.

[GSW19] Ghazimatin, A.; Saha Roy, R.; Weikum, G.: FAIRY: A Framework for Understanding Relationships Between Users’ Actions and their Social Feeds. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019. S. 240–248, 2019, URL: https://doi.org/10.1145/3289600.3290990.

[GW16] Gurevich, Y.; Wing, J. M.: Inverse privacy. Commun. ACM 59/7, S. 38–42, 2016, URL: https://doi.org/10.1145/2838730.

[HKN12] Hijikata, Y.; Kai, Y.; Nishida, S.: The relation between user intervention and user satisfaction for information recommendation. In: Proceedings of the ACM Symposium on Applied Computing, SAC 2012, Riva, Trento, Italy, March 26-30, 2012. ACM, S. 2002–2007, 2012, URL: https://doi.org/10.1145/2245276.2232109.

[KBV09] Koren, Y.; Bell, R. M.; Volinsky, C.: Matrix Factorization Techniques for Recommender Systems. Computer 42/8, S. 30–37, 2009, URL: https://doi.org/10.1109/MC.2009.263.

[Kn12] Knijnenburg, B. P.; Bostandjiev, S.; O'Donovan, J.; Kobsa, A.: Inspectability and control in social recommenders. In: Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012. S. 43–50, 2012, URL: https://doi.org/10.1145/2365952.2365966.

[Mi19] Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artif. Intell. 267/, S. 1–38, 2019, URL: https://doi.org/10.1016/j.artint.2018.07.007.

[NK19] Nikolakopoulos, A. N.; Karypis, G.: RecWalk: Nearly Uncoupled Random Walks for Top-N Recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019. S. 150–158, 2019, URL: https://doi.org/10.1145/3289600.3291016.

[PB15] Parra, D.; Brusilovsky, P.: User-controllable personalization: A case study with SetFusion. Int. J. Hum. Comput. Stud. 78/, S. 43–67, 2015, URL: https://doi.org/10.1016/j.ijhcs.2015.01.007.

[RRS15] Ricci, F.; Rokach, L.; Shapira, B., Hrsg.: Recommender Systems Handbook. Springer, 2015, ISBN: 978-1-4899-7636-9.

[Sa01] Sarwar, B. M.; Karypis, G.; Konstan, J. A.; Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001. S. 285–295, 2001, URL: https://doi.org/10.1145/371920.372071.

[TM07] Tintarev, N.; Masthoff, J.: A Survey of Explanations in Recommender Systems. In: Proceedings of the 23rd International Conference on Data Engineering Workshops, ICDE 2007, 15-20 April 2007, Istanbul, Turkey. S. 801–810, 2007, URL: https://doi.org/10.1109/ICDEW.2007.4401070.

[ZC20] Zhang, Y.; Chen, X.: Explainable Recommendation: A Survey and New Perspectives. Found. Trends Inf. Retr. 14/1, S. 1–101, 2020, URL: https://doi.org/10.1561/1500000066.