Model-Agnostic Counterfactual Explanations of Recommendations

Vassilis Kaffes
"Athena" Research Center
Athens, Greece
vkaffes@athenarc.gr

Dimitris Sacharidis
Université Libre de Bruxelles
Brussels, Belgium
dimitris.sacharidis@ulb.be

Giorgos Giannopoulos
"Athena" Research Center
Athens, Greece
giann@athenarc.gr

ABSTRACT

Explanations for algorithmically generated recommendations is an important requirement for transparent and trustworthy recommender systems. When the internal recommendation model is not inherently interpretable (e.g., most contemporary systems are complex and opaque), or when access to the system is not available (e.g., recommendation as a service), explanations have to be generated post-hoc, i.e., after the system is trained. In this common setting, the standard approach is to provide plausible interpretations of the observed outputs of the system, e.g., by building a simple surrogate model that is inherently interpretable, and explaining that model. This however has several drawbacks. First, such explanations are not truthful, as they are rationalizations of the observed inputs and outputs constructed by another system. Second, there are privacy concerns, as to train a surrogate model, one has to know the interactions from users other than the one who seeks an explanation. Third, such explanations may not be scrutable and actionable, as they typically return weights for items or other users that are difficult to comprehend, and hard to act upon so to improve the quality of one's recommendations.

In this work, we present a model-agnostic explanation mechanism that is truthful, private, scrutable, and actionable. The key idea is to provide counterfactual explanations, defined as those small changes to the user's interaction history that are responsible for observing the recommendation output to be explained. Without access to the internal recommendation model, finding concise counterfactual explanations is a hard search problem. We propose several strategies that seek to efficiently extract concise explanations under constraints. Experimentally, we show that these strategies are more efficient and effective than exhaustive and random search.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

recommender systems, explainability, counterfactuals

ACM Reference Format:
Vassilis Kaffes, Dimitris Sacharidis, and Giorgos Giannopoulos. 2021. Model-Agnostic Counterfactual Explanations of Recommendations. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '21), June 21–25, 2021, Utrecht, Netherlands.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3450613.3456846

1 INTRODUCTION

Recommender systems are becoming ever more prevalent in various aspects of our everyday lives and, at the same time, ever more opaque with complex models trained on a multitude of information sources. The single most important means to increase the transparency of and foster trust in recommenders is via explanations [18, 22], which are succinct, human-oriented pieces of information describing why the system exhibits the observed behavior.

Explainability serves multiple purposes beyond trust and transparency [17]. For example, a system designer may use explanations to fine-tune the model; explanations to a user could help them modify their behavior so as to increase their satisfaction. Naturally, a great amount of research has been devoted to designing *interpretable* recommender systems [1, 2, 8, 23], which by design are easier to explain. However, there are many situations where the need for *explainability* arises but the underlying system is not interpretable and cannot be modified. For example, commercially deployed systems are often carefully optimized to serve the business' internal goals, which do not necessarily align with explainability. Similarly, when recommendations are outsourced and used as a service, the inner workings of the system is an intellectual property of the service provider and cannot be revealed. In situations like these, the recommendations must be explained *post-hoc*, after the system is trained and deployed.

Post-hoc explanation approaches can be distinguished into those that are *model-agnostic* and treat the recommender as a black-box [11, 13], and those that are tied to a specific class of recommendation models [3, 5]. The former have the advantage that they are *universal*, but suffer in *fidelity*, as the explanations are generated with respect to an interpretable model that only approximates the recommender. Conversely, the latter creates truthful explanations but cannot be applied universally. Moreover, all existing methods require access to the training data, raising *privacy* concerns, while some additionally need access to the state of the trained system, which contrasts the recommendation as a service model. Another limitation of some post-hoc explainability methods [3, 11, 13] is that they are not *scrutable*, since the explanations they return are influence weights assigned to items or other users that are difficult to comprehend.

UMAP '21, June 21–25, 2021, Utrecht, Netherlands

ACM ISBN 978-1-4503-8366-0/21/06…$15.00
https://doi.org/10.1145/3450613.3456846

In this paper, we present a post-hoc explanation mechanism that is model-agnostic and does not suffer from the aforementioned limitations. It returns *counterfactual explanations* defined as those minimal changes to the user's interaction history that would result in the system not making the recommendation that is to be explained. Because counterfactuals achieve the desired output on the recommender itself, rather than a proxy, our explanation mechanism has the same fidelity as model-specific post-hoc explanations. Moreover, it is completely private, since no other information besides the user's interaction history is required to extract counterfactuals. Finally, owing to their simplicity, counterfactuals are *scrutable*, as they present specific interactions from the user's history, and potentially actionable (e.g., one requests an explanation for an undesired recommendation, and eventually learns how to change one's inputs to affect the system's outputs).

The search space for finding counterfactual explanations is exponentially large with respect to the number of interactions of a user; any subset of the user's interactions is a candidate explanation. In the absence of any model information, gradient-based optimization methods (such as [10, 20]) cannot be applied. Moreover, without making any assumptions about the recommendation engine (as in [5]), the search space cannot be pruned. Therefore, one has to turn to heuristic strategies to guide the search space towards candidate counterfactuals and avoid an exhaustive enumeration of the search space. We propose several search strategies that exploit information about the behavior of the recommender deduced from its outputs. An exhaustive experimental evaluation shows that our strategies exhibit different performance profiles, and are more efficient that exhaustive search and more effective than random search.

The remaining of the paper is organized as follows. Section 2 overviews related work. Section 3 introduces our approach for generating model-agnostic counterfactual explanations. Then, Section 4 presents an experimental evaluation, while Section 5 discusses conclusions.

2 RELATED WORK

Explanations accompanying recommendations can serve various purposes, either for the user receiving the recommendations, or for the system provider serving them [17]. Explanations can be presented in various modalities to the user, ranging from *textual* information, such as sentences based on templates [23], a list of relevant tags or features [19]; up to more *visual* representations, such as histograms [6], and radar charts [7]. The information to display in explanations may come directly from the training data (e.g., [1, 13]), the recommendation model (e.g., [6, 16]), or external sources (e.g., reviews [4, 9], knowledge graphs [21]).

Research in explainability of recommendations can be divided into two main fields; see also a recent survey [22]. The first is about designing *interpretable recommenders*, i.e., systems intentionally designed to facilitate generating explanations. Note that there are several simple recommenders that are highly interpretable, such as content-based approaches [19], and neighborhood-based collaborative filtering (CF) [6, 16], and thus explanations can be built upon information on users, items, features, or combinations thereof [12], in different modalities [18]. Modern recommenders based on latent factors, embeddings, and deep learning, are not inherently interpretable. In this case, a typical approach is to try to align latent factors with external features or aspects (e.g., extracted from reviews) [2, 8, 23] making them more interpretable. Another approach is to force latent factor models respect user or item neighborhoods (e.g., users that purchased the same item should have similar latent representations) [1], making thus possible to derive explanations similar to neighborhood-based CF.

The second field is concerned with providing *post-hoc explanations*, where the system is not modified and the goal is to explain its outputs *after* it is trained. In this work, we further distinguish between true *black-box* approaches that are *model-agnostic* and are thus not bound to any one specific recommender, and *grey-box* approaches that only apply to specific recommendation models. Grey-box methods typically seek to identify the influence of specific training data points in the recommendation to be explained. To achieve this, the explanation engine must have access to both the internal state of the model and the training data. For example, [3] proposes an efficient method to do influence analysis on the training data. Naively, to extract the influence of each data point, one has to remove it from the training data, retrain the model, and observe the difference in the output. Instead, [3] shows that influence analysis is possible on latent factor models, as long as one has access to the gradients of the loss function. Another approach [5] focuses on recommenders that operate on heterogeneous information networks, i.e., graphs encoding the different interactions (the edges) between users and items (the nodes), and compute a PageRank-like score for the nodes. Similar to our approach, [5] seeks to provide counterfactual explanations that identify those user interactions that when removed would result in a different recommendation. Contrary to our approach, [5] requires access to all training data, the trained model, and of course only works for the specific recommendation engine.

The standard paradigm in model-agnostic (black-box) approaches is to train an interpretable model on the same data. For instance, [13] considers explanations based on association rules, and investigates the trade-off between accuracy and interpretability. A more reasonable approach is to avoid building a single model to explain all instances, and rather build a *local surrogate model* designed to approximate the decisions of the system around the neighborhood of the instance to be explained [15]. The work in [11] applies this idea to explain a rating prediction of a recommender. By carefully selecting a subset of the training data, the method builds a simple linear model that accurately captures the predictions of the recommender. The explanation returned is a set of weights on the user's past rating, indicating their influence on the recommendation. The aforementioned model-agnostic approaches suffer in *fidelity*. Since explanations are generated by a different model than the one making the recommendations, the explanations cannot be truthful and they can only provide with plausible rationalizations of the observed recommender behavior or relationships in the training data.

In general, counterfactual explanations [10, 20] have only been considered in grey-box scenarios, where at least access to the gradients of the model's loss function is assumed [10]. The main reason for this is that the space of possible counterfactual explanations is large, and thus to find the best explanation one has to guide the search via gradient-based optimization. In this work, we show that it is feasible to extract counterfactual explanations in a model-agnostic scenario. Specifically, we guide the search heuristically using information derived from the observed output of the recommender. In some sense, we are using a surrogate model of the recommender, not to create explanations, but to find them. This combines the fidelity of counterfactual explanations, with the minimal requirements imposed by black-box approaches. Moreover, our approach respects the privacy of other users, as it does not require access to all training data.

3 MODEL-AGNOSTIC COUNTERFACTUAL EXPLANATIONS

Problem Definition. In this section, we introduce the notion of model-agnostic counterfactual explanations, and present mechanisms to generate them. Our approach is universal and applies to almost all recommender systems, as we only make the minimum necessary assumptions.

A user has interacted with (purchased, clicked on, viewed, etc.) a set of *interacted items* I, where $n = |I|$. Given as input the set I, the recommender system produces *recommendations* R, which is a ranked list of $m = |R|$ items. The explanation need is the following: the user requests to find out why a particular *target item* $t \in R$ was recommended. We denote as *target position* the position of the target item in the recommendations. For the given request, the explanation mechanism returns a *counterfactual explanation* E, which is a subset of interacted items, $E \subseteq I$, such that had the user only interacted with items $I \setminus E$, the recommender would return a list R' of m recommendations that do not contain the target item, i.e., $t \notin R'$. We additionally term as $C \subseteq I$ a candidate counterfactual explanation, i.e. a subset of items to remove that has not yet been evaluated as to whether it achieves the desired goal of $t \notin R'$. Here, the set of interacted items I has occured and is thus the *factual*, while the set $I \setminus E$ is an alternate but feasible reality, the *counterfactual*. A counterfactual explanation E draws a causal connection [14] between the user's input I and the recommender's output t.

Methodology. As mentioned, the goal of the problem is to search the space of subsets $C \subseteq I$ (candidate solutions) of interacted items in order to identify subset E, which, upon removal from the user interactions, leads to the removal of target item t from the recommendation list. It is evident that the smaller the size of subset E (alternatively the *explanation length*), the better the explanation: a user is expected to more easily understand and be satisfied by a simpler (i.e., shorter) counterfactual explanation [20]. Note that the aforementioned objective is equivalent to maximizing the number of remaining interactions $I \setminus E$. However, searching the whole space of subsets C becomes infeasible as the size n of the interaction history increases, thus heuristics need to be defined for searching this space as effectively and efficiently as possible. Consequently, another restriction of the problem is posed by an available budget B given to perform this search. The budget is defined in terms of the times an algorithm evaluates (i.e., invokes the recommender for) a candidate solution (C) as to whether the goal of removing the target item t is achieved or not.

Thus, a heuristic algorithm needs to balance the objective of discovering a small counterfactual explanation E, with the restriction of spending at most a budget of B. Our search space of candidate solutions C forms a partially ordered set (ordered by inclusion), which can be represented by a lattice (starting from the empty set and ending to I). Upon this lattice, various graph traversing algorithms can be applied in order to identify counterfactuals. Next, we first present two naive traversing solutions and then, three more elaborate algorithms we propose for effectively and efficiently solving the problem.

Baseline Strategies. We first describe the two baseline strategies.

Random Search (Rnd). The strategy randomly considers candidates with respect to both the cardinality and the selected items. It terminates when budget B is spent and keeps, among all counterfactuals (candidates that achieve the target objective), the one E with the minimum size (explanation length) if one exists.

Exhaustive Search (Exh). The strategy considers candidates in increasing cardinality: first it examines all sets with cardinality 1, then with 2, and so forth. The strategy terminates as soon as a counterfactual explanation is found, since this is optimal in terms of its length, or if the budget is spent.

Proposed Strategies. For the following strategies, we introduce two quality measures of a candidate counterfactual C that can guide the search.

The first is the *normalized length*, defined as $l(C) = \frac{|C|}{|I|}$, i.e., the ratio of the number items in the candidate with respect to all interacted items. Because $C \subseteq I$, the normalized length takes values in the range $(0, 1]$.

The second is the *impotence* of a candidate, defined as $i(C) = \max\left\{\frac{m-\mathrm{rank}(t;C)+1}{m}, 0\right\}$, where $\mathrm{rank}(t;C)$ indicates the position of the target item t in the recommendations produced by the system if the set of interacted items was C instead of I. Impotence measures the inability of the set C of interacted items to explain candidate t. On the one hand, if the candidate C is a counterfactual explanation, item t would be ranked after the m-th item, and hence impotence takes its lower value of zero. On the other hand, if C results in item t being ranked top, impotence takes its highest value of one.

For both measures, lower values for a candidate are desired, i.e., small normalized length, and small impotence.

Breadth First Search (BFS). This strategy operates in two phases. In the first, it tries to quickly identify a candidate that is an explanation, while in the second, it seeks to refine that explanation.

Specifically, in its first phase, BFS starts from the empty set and incrementally builds a candidate by appending one item at a time. Let C denote the current candidate. BFS will consider all candidates of one more cardinality, i.e., $C' = C \cup \{i\}$ for every item $i \in I \setminus C$. For each such candidate C' BFS invokes the recommender and based on its response, BFS computes the rank of the target item t and consequently the impotence of C'. The strategy then picks the C' with the lowest impotence and proceed to the next step. This iterative process terminates when a counterfactual explanation is identified.

The second phase begins when a counterfactual explanation, say C, is determined. The goal is to investigate whether there exists any counterfactual that is a subset of C, and is thus shorter. Therefore, BFS initiates a breadth-first search on the sub-lattice rooted at C and containing all subsets of C. Specifically, it investigates all subsets of C with cardinality $|C| - 1$ (at the same lattice level), then those of cardinality $|C| - 2$, and so forth. The strategy terminates if the search is completed or if the budget is depleted. As in all strategies, BFS returns the counterfactual explanation E with the minimum length — which in this case, is the last counterfactual identified (since candidates are considered in decreasing cardinality order).

Priority Search (Pri). This strategy seeks to visit parts of the lattice that appear promising. Each candidate is assigned a priority score computed as a convex combination of its normalized length and impotence:

$$s(C) = \alpha \cdot i(C) + (1 - \alpha) \cdot l(C),$$

where $\alpha \in [0, 1]$ is a weighting factor.

As candidates are explored, they are added to a min-heap with key their priority score. At each iteration of Pri, the candidate C with the least score is deheaped. Since C is the currently most promising candidate, the strategy decides to explore its neighborhood hoping an even more promising candidate would be found. Therefore, Pri considers all subsets of C with cardinality $|C| - 1$ and all supersets of C with cardinality $|C| + 1$. For each candidate in the neighborhood of C, Pri invokes the recommender, computes its priority score, and enheaps it. After the budget is depleted, Pri returns the counterfactual explanation E with the minimum length if one is found.

Hybrid Search (Hyb). This strategy is a hybrid of the exhaustive and the priority search. It is motivated by the fact that in most cases a short explanation can be found. Therefore Hyb chooses to exhaustively consider all candidates up to a fixed small cardinality size, e.g., 2. Obviously, if any counterfactual explanation is found, the strategy terminates. Otherwise, the priority score of each examined candidate is computed, and all candidates are enheaped. From that point on, Hyb behaves exactly like priority search, traversing the lattice until the budget is spent. Compared to Pri, hybrid search essentially bootstraps the priority queue-based search by considering all small-cardinality candidates. In our evaluation, we consider only candidates with up to two interacted items.

4 EXPERIMENTS

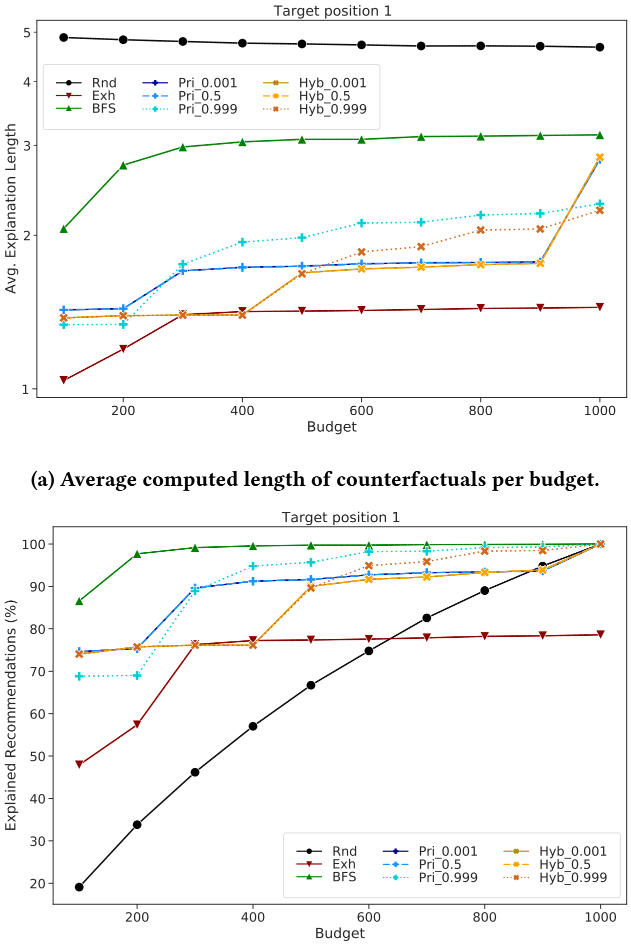
Experimental Setup. We present an experimental evaluation seeking to answer the following question:

Are the proposed heuristic strategies more effective than a simple exhaustive enumeration and a random search for finding counterfactual explanations?

For this reason, we compare our strategies with two baselines, Exhaustive and Random. For the Priority and Hybrid approaches, we consider various weight values α among $\{0.001, 0.5, 0.999\}$.

We consider the following explanation scenario. A user is presented with a recommendation list R, and wishes to receive an explanation about why a specified target item is recommended. The explanation mechanism has a budget to spent looking for a counterfactual explanation. This budget is measured in terms of the number B of calls to the system to generate recommendations; e.g., it is assumed that these calls are expensive in terms of money (paid service) or time (response), or both. When the budget is depleted, the explanation mechanism then returns the best, i.e., shortest, counterfactual explanation it was able to find, or returns "*not explainable*" in case it cannot find one.

As the underlying black-box recommender, we use the LSTM-based session-based recommender implemented by the Spotlight library [1], and the MovieLens 100K dataset. We generate top-20 (i.e., $m = 20$) recommendations for all users with more than 20 interactions in the dataset, and among these interactions, we keep the first 20 interactions so that all requests for explanations have the same characteristics. To measure the *effectiveness* of the strategies, we compute (1) the average length of the returned counterfactual explanation, and (2) the percentage of recommendations that were explained.



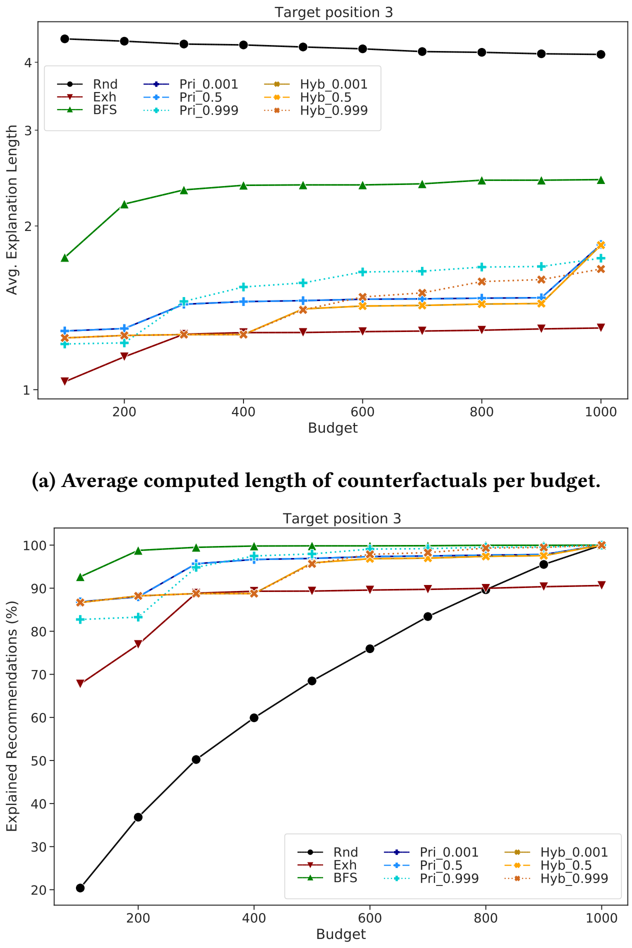
(a) Average computed length of counterfactuals per budget.

(b) Recommendations successfully explained per budget.

Figure 1: Explanation effectiveness of compared methods for target recommendations items at position 1.

Results. In figs. 1a, 2a and 3a, we execute each of the proposed algorithms and utilize a set of specified budgets to spend in order

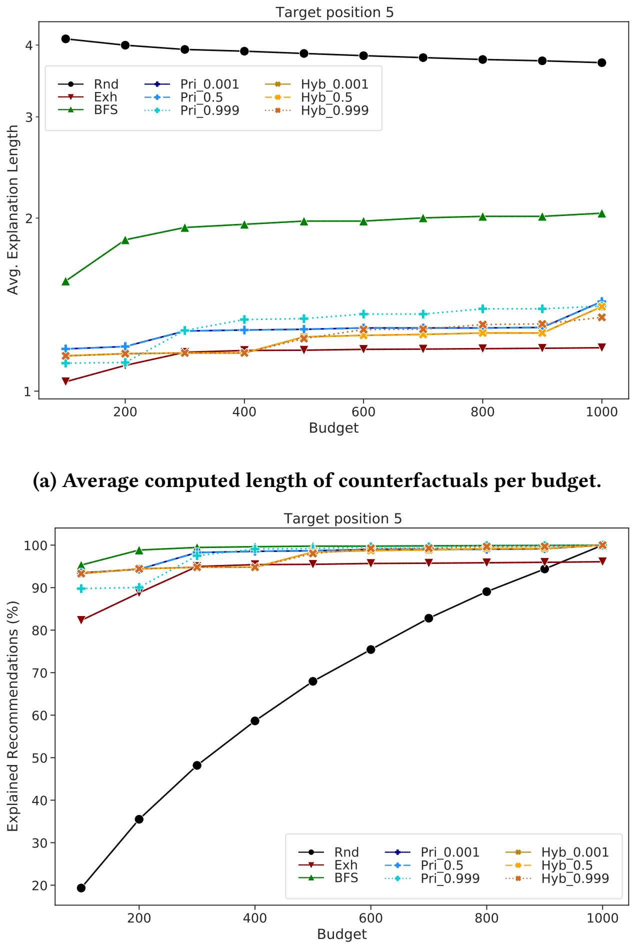
[1] https://github.com/maciejkula/spotlight



(a) Average computed length of counterfactuals per budget.

(b) Recommendations successfully explained per budget.

Figure 2: Explanation effectiveness of compared methods for target recommendations items at position 3.



(a) Average computed length of counterfactuals per budget.

(b) Recommendations successfully explained per budget.

Figure 3: Explanation effectiveness of compared methods for target recommendations items at position 5.

to find a valid solution. For each of these predefined budgets, we calculate the average length of the counterfactual explanations, i.e., sets of removed items that moved the target item beyond the m−th position in the recommendation list within the available budget. Moreover, in figs. 1b, 2b and 3b, we count the number of recommendations that an explanation is provided to the total examined cases, again for a set of specified budgets.

A first comment is that the Random algorithm does not manage to provide satisfying recommendations, neither regarding the size of the counterfactual, nor w.r.t. to handling all the cases, especially when the budget is relatively small, as expected. On the other hand, BFS and Exhaustive give the upper and lower limit of the best solutions anyone could expect taking into account different aspects of the problem. BFS provides a fast solution without spending a lot of the available budget at the expense of very long counterfactuals in length, i.e., average explanation length is high. Exhaustive finds out the shortest counterfactual, in length terms. However, the explained recommended cases are lower that the rest of the proposed algorithms, except for Random in most examined budgets. Although none of the algorithms seem to provide the best overall solution, Priority and Hybrid, with $\alpha = 0.999$ weight, manage to give good solutions. The Priority algorithm explains more recommendations than Hybrid at the expense of a higher average explanation length. Finally, we notice that searching for good recommendations is highly affected by the position of the target item in the recommendation list. As the target position is closer to the m−th position of the list, i.e., figs. 3a and 3b, both examined metrics are getting better and this affects all the algorithms.

Findings. The following conclusions can be drawn from the evaluation. Regarding the baselines, Random Search can identify solutions, but requires considerable budget in order to return short explanations. Exhaustive Search can identify counterfactuals that are short, but soon depletes the budget and cannot provide explanations for the hard cases, where more than three interactions (i.e. $|E| > 3$)

have to be removed. The BFS method is very good at finding valid explanations, exhibiting the highest rate of explained recommendations at a low budget, but at the cost of creating relatively long explanations that may be harder to interpret. So if providing explanations to as many users as possible is of primary concern, then BFS should be preferred. The Priority and the Hybrid search strategies provide a better balance between the quality of the explanation, its length, and the time they spent looking for one. Specifically, Hybrid is able to both derive early on good recommendations, and improve upon the initial explanations over time.

5 CONCLUSION

This work presented a model-agnostic post-hoc explanation mechanism for recommendations that is truthful, private, scrutable, and actionable. From the user's interactions history, it extracts counterfactuals, which are those interactions that when removed would result in a different recommendation. In the absence of any information about the recommender, we propose several heuristic search strategies that are able to quickly identify succinct explanations.

ACKNOWLEDGMENTS

This research has been partially supported by the EU project XMANAI (H2020-ICT-38-2020-957362).

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 2017, Como, Italy, August 27-31, 2017*, Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, and Alexander Tuzhilin (Eds.). ACM, 79–83. https://doi.org/10.1145/3109859.3109913

[2] 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, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 717–725. https://doi.org/10.1145/3097983.3098170

[3] Weiyu Cheng, Yanyan Shen, Linpeng Huang, and Yanmin Zhu. 2019. Incorporating Interpretability into Latent Factor Models via Fast Influence Analysis. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 885–893. https://doi.org/10.1145/3292500.3330857

[4] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan S. Kankanhalli. 2018. Aspect-Aware Latent Factor Model: Rating Prediction with Ratings and Reviews. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien L. Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 639–648. https://doi.org/10.1145/3178876.3186145

[5] Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. 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*, James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 196–204. https://doi.org/10.1145/3336191.3371824

[6] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *CSCW*.

[7] Yunfeng Hou, Ning Yang, Yi Wu, and Philip S. Yu. 2019. Explainable recommendation with fusion of aspect information. *World Wide Web* 22, 1 (2019), 221–240. https://doi.org/10.1007/s11280-018-0558-1

[8] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Coevolutionary Recommendation Model: Mutual Learning between Ratings and Reviews. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien L. Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 773–782. https://doi.org/10.1145/3178876.3186158

[9] Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, Qiang Yang, Irwin King, Qing Li, Pearl Pu, and George Karypis (Eds.). ACM, 165–172. https://doi.org/10.1145/2507157.2507163

[10] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, Mireille Hildebrandt, Carlos Castillo, Elisa Celis, Salvatore Ruggieri, Linnet Taylor, and Gabriela Zanfir-Fortuna (Eds.). ACM, 607–617. https://doi.org/10.1145/3351095.3372850

[11] Caio Nóbrega and Leandro Balby Marinho. 2019. Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*, Chih-Cheng Hung and George A. Papadopoulos (Eds.). ACM, 1671–1678. https://doi.org/10.1145/3297280.3297443

[12] Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. 2012. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Min. Knowl. Discov.* 24, 3 (2012), 555–583. https://doi.org/10.1007/s10618-011-0215-0

[13] Georgina Peake and Jun Wang. 2018. Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 2060–2069. https://doi.org/10.1145/3219819.3220072

[14] Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect.* Basic books.

[15] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *KDD*.

[16] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. 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*, Vincent Y. Shen, Nobuo Saito, Michael R. Lyu, and Mary Ellen Zurko (Eds.). ACM, 285–295. https://doi.org/10.1145/371920.372071

[17] Nava Tintarev and Judith Masthoff. 2007. A Survey of Explanations in Recommender Systems. In *ICDEW*.

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

[19] 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 2009, Sanibel Island, Florida, USA, February 8-11, 2009*, Cristina Conati, Mathias Bauer, Nuria Oliver, and Daniel S. Weld (Eds.). ACM, 47–56. https://doi.org/10.1145/1502650.1502661

[20] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *CoRR* abs/1711.00399 (2017).

[21] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 417–426. https://doi.org/10.1145/3269206.3271739

[22] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Found. Trends Inf. Retr.* 14, 1 (2020), 1–101. https://doi.org/10.1561/1500000066

[23] 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 *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 83–92. https://doi.org/10.1145/2600428.2609579