Recommendation Recovery with Adaptive Filter for Recommender Systems

José Miguel Blanco[1] [a], Mouzhi Ge[2] and Tomáš Pitner[1]
[1]Faculty of Informatics, Masaryk University, Brno, Czech Republic
[2]Deggendorf Institute of Technology, Germany

Keywords: Recommender Systems, Recommendation Recovery, Adaptive Filter, User-oriented Recommendation.

Abstract: Most recommender systems are focused on suggesting the optimal recommendations rather than finding a way to recover from a failed recommendation. Thus, when a failed recommendation appears several times, users may abandon to use a recommender system by considering that the system does not take her preference into account. One of the reasons is that when a user does not like a recommendation, this preference cannot be instantly captured by the recommender learning model, since the learning model cannot be constantly updated. Although this can be to some extent alleviated by critique-based algorithms, fine tuning the preference is not capable of fully expelling not-preferred items. This paper is therefore to propose a recommender recovery solution with an adaptive filter to deal with the failed recommendations while keeping the user engagement and, in turn, allow the recommender system to become a long-term application. It can also avoid the cost of constantly updating the recommender learning model.

1 INTRODUCTION

Recommender Systems (RS) play an important role in our life, from recommending the next favourite movie or song to suggesting a particular fitness exercise (Chedrawy and Abidi, 2009; Lavanya et al., 2021; Sanchez et al., 2020). But their use has expanded beyond those common uses to reach the point on which their use has been linked to more difficult questions as where to travel (Mahmood et al., 2008). Their implementation has gotten to the point where they aim to have a real conversation with the user (Jannach et al., 2020).

Most have assumed that ensuring a higher satisfaction from the user is achieved by making the best possible prediction (Fang et al., 2020). However, once RS fail to recommend the optimal item, they may neglect considering the reasons for the unsuccessful recommendation. Thus, this may lead to a dichotomy in which a set of users is left behind. To alleviate this issue, explanations can be used to improve RS (Naiseh et al., 2020) but it does not address the failure itself. Also, it can be partly attributed to the reason that RS are considered to be used as one-shot services rather than long-term services (Mimoun et al., 2017). Therefore, for RS, it is not only important to predict the best possible recommendation, but also is critical to recover from the failed recommendations.

Therefore, catching the user preference has become a hot topic in the field. While analyzing the reasons for failure (Ben Mimoun et al., 2012) might help to understand the next step to be taken, there are other approaches focused on obtaining direct results. For example (Narducci et al., 2018) has focused on using conversational RS to improve the user experience. Similarly, (Kang et al., 2017) explains how the users utilize natural language to look for a recommendation, thus being able to flag certain behaviours that point directly to catching their preference. It is worth noting, that to catch user preference, a point in which their personalities are modelled has been reached (Alves et al., 2020).

In order to tackle the recommendation recovery, the main aim of this paper is to propose a solution for recovering failed recommendations, i. e., when RS recommend a suboptimal item, it can be used to address the user un-satisfaction for recommendations. Further, we aim to make the RS sustainable for a longer period of time without increasing the computational complexity of it or having to deal with a cold start again (Han et al., 2019). This solution is driven

[a] https://orcid.org/0000-0001-9460-8540

Blanco, J., Ge, M. and Pitner, T.
Recommendation Recovery with Adaptive Filter for Recommender Systems.
DOI: 10.5220/0010653600003058
In *Proceedings of the 17th International Conference on Web Information Systems and Technologies (WEBIST 2021)*, pages 283-290
ISBN: 978-989-758-536-4; ISSN: 2184-3252

by an *adaptive filter*, which after a recommendation failure, filters out all items that are similar to the one disliked by the user. We also validate the solution in the healthcare RS domain.

The structure of the paper is organized as follows: Section 2 reviews the state-of-the-art works that are related to recommendation refinement and recovery. Based on the review, Section 3 introduces a series of definitions and technical notions that help to understand the adaptive filtering technique that we aim to implement. We also include the process model of how the proposed adaptive filter should work with a visual representation of the solution. Section 4 is dedicated to the preliminary validation in which we show how a RS is prepared to fail and discard suboptimal choices in the concrete case of the use of medical drugs in the Emergency Room. In Section 5 we discuss the proposed solution by comparing with critique-based RS and highlighting the novelty of the work. Finally, Section 6 concludes the paper and outlines future research work.

2 RELATED WORKS

There are several research works about RS that try to improve the user experience. Most of those works are to find a characteristic related to the recommendation and use it as the main pillar to improve recommendations. Those recommenders can be classified as characteristic-based fine tuning, boost factor improvement, and interaction-based recovery.

Previous papers such as (Calero Valdez et al., 2016), (Sidana et al., 2021), (Burke,), (Ziegler, 2005) and (Codina and Ceccaroni, 2010) show that there are different valid technical approaches when developing a RS. This is related to how the novelty of excluding the items that our approach proposes is worth developing. (Ziegler, 2005) and (Codina and Ceccaroni, 2010) are focused on how RS works within the framework that the semantic web provides. While the first is a compendium of all the techniques that can be applied, the latter is a direct application of given techniques. Also, (Sidana et al., 2021) provides a new framework for collaborative filtering RS for a smaller user-loss despite a minimal choosing of elements. The work is further expanded into a Neural-Network model to support the analysis of the data. Another type of framework is to fine tune the recommendation. For example, (Calero Valdez et al., 2016) shows the use of RS in Health Informatics. They focus on explaining how the use of a doctor-in-the-loop figure would help the systems to fine tune the system and provide a more clear recommendation. They also propose a framework for evaluating this kind of RS. To mix the frameworks and achieve high quality recommendation, (Burke,) shows how hybrid web RS, those that use multiple approaches instead of just one when generating a recommendation, are better suited than those that are not. This leads to a higher satisfaction of the user.

The works from (De Pessemier et al., 2010) and (Mendoza and Torres, 2020) show how the best possible recommendation is generated based on the result of certain boost factor in the RS. In (De Pessemier et al., 2010), the authors focus on showing how time affects the quality of data collected in collaborative algorithms to provide the end user with the best recommendation possible. On the other hand, (Mendoza and Torres, 2020) introduces a framework on which the bias for popular items is alleviated by introducing an evaluation tool on new items regarding their novelty with respect to the characteristics of the most popular items.

Finally, most of the work on RS is based on how to process the data generated by the user profiles. Papers are focused on obtaining the best recommendation *a priori*, even though there are exceptions as (Li et al., 2018). These papers show the importance of listening to the choice that the user is making and being able to respond afterwards. For example, (Adomavicius et al., 2011) goes in detail into Context Aware RS that make use of the context generated data for the user. The position that the authors hold is that including all the context data the recommendation will be more tailored for a specific user. (Li et al., 2018) develops a method focused on group recommendation and interactive preference. They include a mechanism to generate feedback from a post-rating system. They evaluate the work done by comparing with traditional collaborative filtering. (Vajjhala et al., 2021) built a RS based on the analysis of the user Twitter's profile, therefore recommending items and services catered to their tastes. They are able to show a strong correlation between the user's tweets and the category of items/services the user consumes. (Jin et al., 2020) focuses on showing how the characteristics of the users, namely *visual memory* and *musical sophistication* help the RS to provide a much better recommendation. They found that modifying the design could help to get better acceptance from the user while the *musical sophistication* can play a role against the recommendation. (Narang et al., 2021) aims to combine multiple data from the characteristics of the user in an interpretable manner to obtain a much better recommendation. The model proposed offers a perspective on the importance of each characteristic depending on which dataset is being used.

(Ayub et al., 2020) propose an unified approach on which explicit trust, implicit trust and user preference similarity get unified in a rating profile for the target user. This leads to produce more accurate recommendation. (Nalmpantis and Tjortjis, 2017) shows a RS in which personality tests are combined with a pre-existing movie RS. This leads to an increase in the satisfaction of the user when comparing with current and available RS. (Tian et al., 2021) proposes a framework that takes into consideration the order on which the user has interacted with different items. It reduces the list of all the items to just the most relevant, making the choosing of the user more simple. The experiments show how beneficial this approach is. Those works are specially critical for our case, as we are focusing on how the RS responds to the interaction. (Prasad, 2005) intends to solve the "sequence recognition problem", the situation on which the RS is unable to process the fact that the user already has purchased items more advanced than those that are being recommended. The proposed solution is based on a hybrid model that reinforces a collaborative filtering with a case-based reasoning tool for e-commerce. We group the related works according to their interests as shown in Table 1.

Table 1: Recovering user experience in recommender systems.

Characteristic-based fine tuning
(Ziegler, 2005)
(Codina and Ceccaroni, 2010)
(Sidana et al., 2021)
(Calero Valdez et al., 2016)
(Burke,)
Boost factor improvement
(De Pessemier et al., 2010)
(Mendoza and Torres, 2020)
Interaction-based recovery
(Adomavicius et al., 2011)
(Li et al., 2018)
(Vajjhala et al., 2021)
(Jin et al., 2020)
(Narang et al., 2021)
(Ayub et al., 2020)
(Nalmpantis and Tjortjis, 2017)
(Tian et al., 2021)
(Prasad, 2005)

3 MODEL OF ADAPTIVE FILTER

As we have seen in Section 2, there is a need for newer approaches on how to interpret user interaction and her satisfaction. Also, the lack of references on how to act when a recommendation has failed propels even more the proposal of this adaptive filter.

In this section we introduce the technical definitions of the solution we are presenting. First of all, we define the set of items to be recommended and the set of users that are getting recommended an item:

Definition 3.1 (Items and their Set)**.** I is a non-empty and non-trivial set such that $I = \{i_1, i_2, i_3, ..., i_m\}$. Each i_j represents a different item; therefore, I should be regarded as the set of items to be recommended. Additionally, for any item i_j, it is built as follows: $i_j = \{c_1, c_2, c_3, ..., c_l\}$. Each c_h represents a different characteristic of the item i_j.

Definition 3.2 (Set of Users)**.** $\mathcal{U}$ is a non-empty and non-trivial set such that $\mathcal{U} = \{u_1, u_2, u_3, ..., u_n\}$. Each u_k represents a different user; therefore, $\mathcal{U}$ should be regarded as the set of users that are getting recommended an item.

Now we define the core notion of a RS upon which will define the adaptive filter for suboptimal recommendations.

Definition 3.3 (Core Recommender System)**.** A core RS (cRS) is a function that for each user u_k in $\mathcal{U}$, orders the elements of I from 1 to n, where $n = |I|$, according to a certain criteria. The criterion on which the items are ranked is dependant on how the RS is set-up. For example, in a neighborhood RS, the items would be ranked according to how other users have evaluated those items. Nevertheless, given our intention to make the results as universal as possible, these criteria can be any one that the reader might like. Therefore, the set I is transformed into I_r, the ranked set of items. We define I_r as a well-ordered set, the result of applying an ordering operation (the base algorithm of the RS), on I. Afterwards, u_k is recommended the first element of the ordered set, I_r. By $R(u_k, i_j)$ we mean that the user u_k is recommended the item i_j. Furthermore, by $i_j = q$ we mean that the item i_j is in the position q in the set of ranked items. All of the above can be expressed as follows:

$\mathbf{cRS} = \forall \mathbf{u_k}, \mathbf{u_k} \in \mathcal{U}, \exists \mathbf{i_j}, \mathbf{i_j} \in I_r, \mathbf{i_j} = \mathbf{1}$, and $\mathbf{R}(\mathbf{u_k}, \mathbf{i_j})$

It is important to take into account that the proposed definition of a cRS is done with an RS that only recommends the first item in mind. Nevertheless, it could be easily modified so that the cRS actually recommends a subset of items from I_r. In the case we are working with more than one item, the user would be recommended the items ordered from the 1st position to pth position, where p is the number of items to be recommended.

After we have defined the notion of cRS, we need to define how the evaluation of an item happens and the similarity between items.

Definition 3.4 (Evaluation of an Item)**.** $E(u_k, i_j)$ means that the user u_k evaluates the item i_j. This has two possible outcomes: (1) $E(u_k, i_j) = Sat$ and (2) $E(u_k, i_j) = \neg Sat$. If the result of the evaluation is (1), the RS has succeeded in recommending an item; i. e., the user is satisfied and the recommended item was optimal. If the result is (2), we say that the RS has failed; i. e., the user is not satisfied and the recommended item was suboptimal.

Definition 3.5 (Similar Items)**.** Any two items i_j and i_l are similar, in symbols $S(i_j, i_l)$, if and only if they share most of their characteristics.

After introducing the previous concepts, we define the adaptive filter.

Definition 3.6 (Adaptive Filter)**.** An adaptive filter is a cRS in which, after an evaluation such that $E(u_k, i_j) = \neg sat$, the set I_r is revised via filtering. For this, the set I_r is modified so that all the items that are similar to the suboptimal item i_j are pulled out, thus giving birth to a new ranked set of items I_{r2}. As the set I_{r2} is built, so is the set I_d, the set of discarded items. This set is built as follows:

$$I_\mathbf{d} = \{\forall \mathbf{i_n} | \mathbf{S}(\mathbf{i_j}, \mathbf{i_n}), I_\mathbf{d} \cup \{\mathbf{i_n}\}\}$$

Then, for any user u_k and a selected item i_j, the new ranked set I_{r2} is built from the original set I_r as follows:

$$I_\mathbf{r2} = \{\forall \mathbf{i_n} | \mathbf{S}(\mathbf{i_j}, \mathbf{i_n}), I_\mathbf{r} - \{\mathbf{i_n}\}\}$$

This can be iterated a finite number of times p at most, where $p = |I_r|$. Also, it is obvious that I_{r2} can be defined from I_r and I_d, but we have included its definition as standalone to make everything clearer. From an implementation point of view it should be defined from both sets so it is less taxating for the system.

Definition 3.7 (Similarity Threshold)**.** Definition 5 is an obvious application of the Jaccard Index (Lee, 2017) and, therefore, we need to establish a Similarity Threshold (ST). This ST is equal to the ratio of the summatory of the constants of preference (k) of each characteristic, per item, multiplied by the fraction of the set of discarded items. Thus, ST is as follows:

$$\mathbf{ST} = \left(\frac{\sum \mathbf{k_{c_1}}, \ldots, \mathbf{k_{c_n}}}{|I|}\right) \times \left(\frac{\mathbf{1}}{|I_\mathbf{d}|}\right)$$

For qualitative characteristics, the previous applies automatically. For quantitative characteristics, they are to be considered equal if and only if, for characteristics c_f and c_g, $c_f = c_g \pm 15\%$ follows. All of the above means that whenever an RS fails consecutively, less and less items are pulled from the set. Therefore, even if the RS fails too many times consecutively, the set I_r would not be emptied.

Remark 3.1 (Constant of Preference). The constants of preference (k) are introduced in the previous definition to ponder the characteristics to each user. These constants of preference are to be obtained from the user in the same way the RS would deal with a cold start (Han et al., 2019). Some of the tools that can be used for that matter include small surveys at the beginning of the use of the RS, or emotion detection on the user's previous reviews (Ishwarya et al., 2019) among others. The reader might choose the one that feels more adequate.

3.1 Adaptive Filtering Process

After defining the adaptive filter, we show its functionality more clearly with a process model. For that matter, Figure 1 is the representation of a comparison of the process model of how an adaptive filter works against a regular recommender system without the adaptive filter.

A starting set of items I is passed through the preferences of the user u thus, obtaining a ranked set of items I_r. This ranked set, I_r, allows us to present an item i_1 as a recommendation to the user u. After the user makes the purchase of the recommended item, there are two different possibilities: either u is satisfied with the item, $E(u, i_1) = Sat$, and so the ranked set of items is just missing the previously recommended item; or either the user is not satisfied with

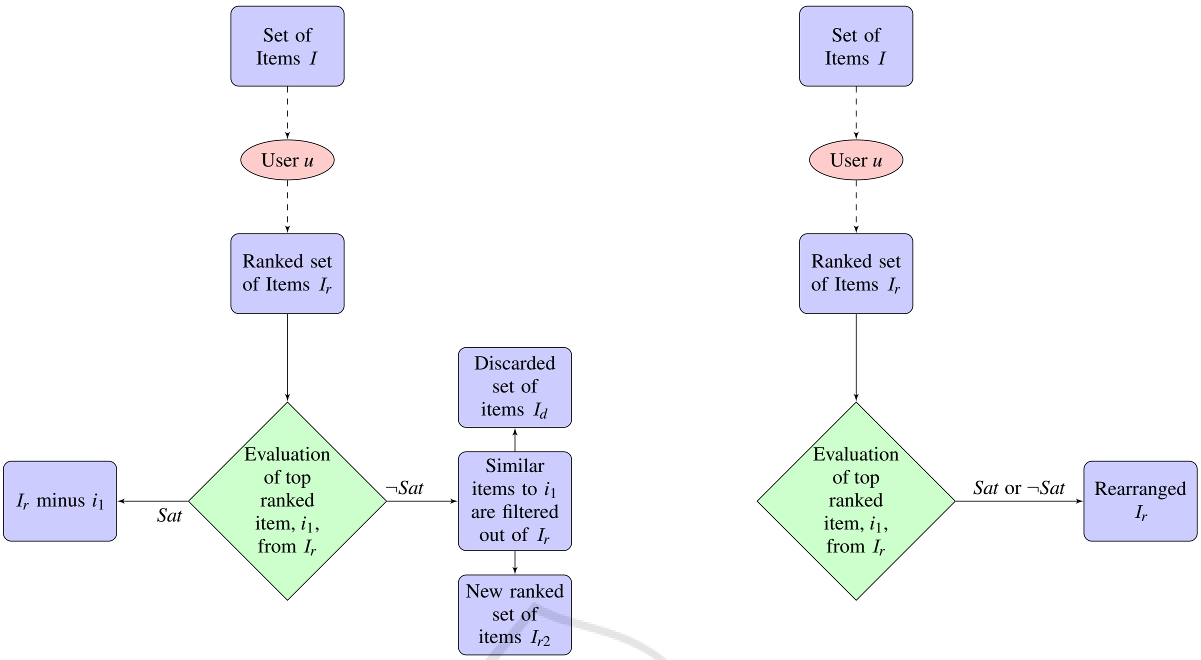

Figure 1: Adaptive Filter Process Model (l) and Recommender System without the Adaptive Filter Process Model (r).

the item, $E(u, i_1) = \neg Sat$, and two new sets are created: a set of discarded items, I_d, containing all the items from the original ranked set that were similar to the non-satisfactory item, and another, I_{r2}, that is obtained subtracting the elements of the set of discarded items from the original set of items I.

4 PRELIMINARY VALIDATION

To further describe the implementation of an adaptive filter, we describe a validation from a within-subject study design perspective. This is due to the fact that it allows for a quicker and better identification of differences rather than a between-subject design. We are validating two different scenarios based on a fragment of a real-world dataset (U. S. Food & Drug Administration, 2021). In both scenarios, a patient has reached the Emergency Room with an unspecific disease. After a preliminary diagnosis the medical team is using two different systems that allow them to rate and find the perfect medical drug to be administered. Both systems can be inputted if the treatment worked or not. Both systems have a built-in medical drug RS, that is selecting which the recommendation is, takes care of the symptoms and compares other people cases. Each time the systems recommend just one medical drug to be used, this is supported by the concern that to obtain a better diagnosis just one drug can be used at the same time. Furthermore, drug use usually needs to be carefully planned and one usually does not work on various options at the same time. In the second scenario, the medical team uses a system whose RS has the adaptive filter technique from Definition 6. The fragment of interest of the set of drugs that the systems are using is the following:

DrugsDataset={Ashlyna, Daysee, Jaimiess, Malmorede, Namenda, Namzarinc, Prozac, Sarafem, Zovia}

Each of these drugs has some characteristics, as instantiated from Definition 1. These characteristics are their active ingredient, strength and route form. Furthermore, each one of them shall be used in obtaining a recommendation. These characteristics have been selected as basic elements to make a drug itself: the active ingredient represents what the drug focuses on and, therefore, what it would be useful for; the strength is a clear indicator of the performance; and the route is sign of how easy or hard would be to apply that drug. Nevertheless, it is worth mentioning that these characteristics are just a representation of a much bigger set and the reader might feel compelled to choose different ones for a different validation. All this is summarized in Table 2.

The medical team has administered Malmorede and Ashlyna; the patient has a good response to those. Given that the patient is improving with the

Table 2: Dataset Fragment of Medical Drugs.

Drug Name	Active Ingredient	Strength	Route
Ashlyna	Ethinyl Estradiol; Levonorgestrel	0.03mg; 0.01mg	Tablet; Oral
Daysee	Ethinyl Estradiol; Levonorgestrel	0.03mg; 0.02mg	Tablet; Oral
Jaimiess	Ethinyl Estradiol; Levonorgestrel	0.02mg; 0.01mg	Tablet; Oral
Malmorede	Ethinyl Estradiol; Ethynodiol Dyacetate	0.05mg; 1mg	Tablet; Oral
Namenda	Memantine Hydrochloride	5mg	Tablet; Oral
Namzaric	Donepezil Hydrochloride; Memantine Hydrochloride	10mg; 14mg	Capsule; Oral
Prozac	Fluoxetine Hydrochloride	10mg	Capsule; Oral
Sarafem	Fluoxetine Hydrochloride	20mg	Capsule; Oral
Zovia	Ethinyl Estradiol; Ethynodiol Dyacetate	0.05mg; 1mg	Tablet; Oral

drugs administered, both systems have ordered the Drugs Dataset as follows:

$DrugsDataset_r$={Daysee=1, Jaiminess=2, Zovia=3, Prozac=4, Namenda=5, Sarafem=6}

Thus, the medical team is recommended to continue the treatment with Daysee, as it comes from Definition 3. The main reason is that it shares active ingredient and has a similar strength to the last drug used, Ashlyna. Trusting the system of each scenario, Daysee is administered. However, in this case, the drug has little to no effect, and so the medical team evaluates the drug as in Definition 4. In the first scenario, after letting the system know that the drug was not cutting it, the recommendation is administering Jaiminess, the next drug highest in the ranked set. With this drug being quite similar to Ashlyna and Daysee, there is a high probability that the drug will not work properly, therefore, the medical team will stop being satisfied with the RS. In the second scenario, the one with a system with a RS with the adaptive filter, after letting the system know that the drug was not efficient, the adaptive filter built in the system filters out drugs with similar characteristics, as it follows from Definition 6. In this case, as Jaimiess has the same active ingredient, route and the strength is in a 15% range of variation, as specified in Definition 7. The newly ranked set of drugs is as follows:

$DrugsDataset_{r2}$={Zovia=3, Prozac=4, Namenda=5, Sarafem=6}

Also, a new set of discarded drugs is built:

DiscardedDrugs={Jaiminess}

These new sets are built according to each of the algorithms of Definition 6. Finally, it recommends them to use to Zovia, the best ranked Drug in the updated set of drugs. It can be seen that adaptive filter offers a viable alternative for a user who is facing a suboptimal recommendation. In this example, the medical team, after being given a suboptimal recommendation, Daysee, is offered a new option, Zovia, as the best fit. This is because Zovia shares most of its characteristics with the previous experiences from the team. Furthermore, they may trust in the adaptive filter RS for further recommendations, as it considers recommendation recovery. Therefore, the life-span of the RS has gone from a one-shot to a multiple-use, and may become a crucial part of the medical team's diagnostic cycle. We can further infer that the satisfaction of the user may improve, even if there has been a suboptimal recommendation, as they are offered a new alternative after a bad experience. But this alternative has been catered to their previous experiences. Additionally, some drugs are pulled out of the drugs dataset and they will not see them recommended again for the same clinical case. This serves to improve their confidence in the agent and extend their use of the RS, since they may feel that their preferences are taken into account.

5 DISCUSSION

One of the effective solutions to address the unsatisfied recommendations is the critique-based RS (Jannach and Zanker, 2020). We name our solution as adaptive filter RS and compare with critique-based RS. Both solution are user-oriented and their core techniques might rely in the same intuitions: that they intend to let users instantly adjust the recommendations in the case that the recommended items are not favored by the users.

There are several differences between the two solutions. The main one is the intention behind each system: while critique-based systems are focused on fine-tuning the users preferences, the adaptive filtering focuses on avoiding any recommendation pitfalls the user might find. Critique-based RS focus on fine tuning the best options available (Ramnani et al., 2018), whereas the adaptive filter RS attempt to hide the worst recommendations.

On the technical aspect, adaptive filtering happens **after** the purchase, while critique-based systems are set **before** or **after** the purchase. The purchase does not make a difference for the latter. Additionally, critique-based RS can only flag items as low-priority, while adaptive filter expels them out of the recommendation pool. It is thus possible that critique-based RS may recommend the same items again, even if they are discarded by the user. On the other hand, items discarded by an adaptive filter are not shown again, therefore finding a working niche in which the user does not want to be shown similar items ever again. Also, it is worth mentioning that our solution, as it decreases the set of items to be recommended each time when the RS fails, makes the computational and interactive costs also decrease.

Finally, as we have seen in Section 4, the value that the adaptive filter RS can offer to some specific domains is immense. In the particular case of healthcare, the proposed solution offers an efficient method to avoid recommending certain items that are of limited usage, which makes the system more intuitive and when applied in healthcare domain, it might be crucial to save a life. This is particularly important in situations of the Emergency Room, the proposed solution is able to exclude a whole family of medical drugs that has already shown limited effects on the patient. This can further facilitate the doctors to conduct more accurate diagnosis and treatment.

6 CONCLUSION

In this paper, we have proposed an adaptive filter for recommender system that is designed to recover failed recommendations. It can be easily integrated to the existing RS without modifying the recommendation algorithm, and the proposed adaptive filter technique adds limited computational complexity. The additional computational complexity is the creation of the set of discarded items. In order to validate the proposed solution, we have conducted a preliminary evaluation with the medical recommender systems. The evaluation result has showed that the proposed solution offers an efficient method to avoid recommending certain items that are of limited effects on patients. We believe that the adaptive filter can be further applied in other domains to recover the failed recommendations.

To continue this research, there are multiple lines of further investigation that are to be explored in the future. Among others, we will include the definition of tools that allow for an evaluation of RS that implement this technique, and integrate this solution into existing RS. The development of evaluation tools would allow to quantify the performance and the improvement that the adaptive filter offers when compared to traditional RS, something that cannot be done in the time being. Furthermore, it would be interesting to see its integration with semantic web RS as those of (Ziegler, 2005) but from the perspective of inconsistent data offered on (Blanco et al., 2021).

REFERENCES

Adomavicius, G., Mobasher, B., Ricci, F., and Tuzhilin, A. (2011). Context-Aware Recommender Systems. *AI Magazine*, 32(3):67–80. Number: 3.

Alves, P., Saraiva, P., Carneiro, J., Campos, P., Martins, H., Novais, P., and Marreiros, G. (2020). Modeling Tourists' Personality in Recommender Systems: How Does Personality Influence Preferences for Tourist Attractions? In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pages 4–13, New York, NY, USA. Association for Computing Machinery.

Ayub, M., Ghazanfar, M. A., Mehmood, Z., Alyoubi, K. H., and Alfakeeh, A. S. (2020). Unifying user similarity and social trust to generate powerful recommendations for smart cities using collaborating filtering-based recommender systems. *Soft Computing*, 24(15):11071–11094.

Ben Mimoun, M. S., Poncin, I., and Garnier, M. (2012). Case study—Embodied virtual agents: An analysis on reasons for failure. *Journal of Retailing and Consumer Services*, 19(6):605–612.

Blanco, J. M., Ge, M., and Pitner, T. (2021). Modelling inconsistent data for reasoners in web of things (accepted). In *Procedia Computer Science*. Elsevier.

Burke, R. Hybrid Web Recommender Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, pages 377–408. Springer, Berlin, Heidelberg.

Calero Valdez, A., Ziefle, M., Verbert, K., Felfernig, A., and Holzinger, A. (2016). Recommender Systems for Health Informatics: State-of-the-Art and Future Perspectives. In *Machine Learning for Health Informatics: State-of-the-Art and Future Challenges*. Springer International Publishing.

Chedrawy, Z. and Abidi, S. S. R. (2009). A Web Recommender System for Recommending, Predicting and Personalizing Music Playlists. In *Web Information Systems Engineering - WISE 2009*, pages 335–342, Berlin, Heidelberg. Springer.

Codina, V. and Ceccaroni, L. (2010). A Recommendation System for the Semantic Web. In *Distributed Comput-*

ing and Artificial Intelligence, Advances in Intelligent and Soft Computing, Berlin, Heidelberg. Springer.

De Pessemier, T., Dooms, S., Deryckere, T., and Martens, L. (2010). Time dependency of data quality for collaborative filtering algorithms. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 281–284, New York, NY, USA. Association for Computing Machinery.

Fang, H., Zhang, D., Shu, Y., and Guo, G. (2020). Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations. arXiv: 1905.01997.

Han, D., Li, J., Li, W., Liu, R., and Chen, H. (2019). An app usage recommender system: improving prediction accuracy for both warm and cold start users. *Multimedia Systems*, 25(6):603–616.

Ishwarya, M. V., Swetha, G., Saptha Maaleekaa, S., and Anu Grahaa, R. (2019). Efficient Recommender System by Implicit Emotion Prediction. In *Advances in Big Data and Cloud Computing*, Advances in Intelligent Systems and Computing, pages 173–178.

Jannach, D., Manzoor, A., Cai, W., and Chen, L. (2020). A Survey on Conversational Recommender Systems. *arXiv:2004.00646 [cs]*. arXiv: 2004.00646.

Jannach, D. and Zanker, M. (2020). Interactive and Context-Aware Systems in Tourism. In *Handbook of e-Tourism*, pages 1–22. Springer International Publishing, Cham.

Jin, Y., Tintarev, N., Htun, N. N., and Verbert, K. (2020). Effects of personal characteristics in control-oriented user interfaces for music recommender systems. *User Modeling and User-Adapted Interaction*, 30(2):199–249.

Kang, J., Condiff, K., Chang, S., Konstan, J. A., Terveen, L., and Harper, F. M. (2017). Understanding How People Use Natural Language to Ask for Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 229–237, New York, USA. ACM.

Lavanya, R., Khokle, T., and Maity, A. (2021). Review on Hybrid Recommender System for Mobile Devices. In *Artificial Intelligence Techniques for Advanced Computing Applications*, pages 477–486, Singapore.

Lee, S. (2017). Improving Jaccard Index for Measuring Similarity in Collaborative Filtering. In *Information Science and Applications 2017*.

Li, B.-H., Zhang, A.-M., Zheng, W., Wan, S., Qin, X.-L., Li, X., and Yin, H.-L. (2018). GRIP: A Group Recommender Based on Interactive Preference Model. *Journal of Computer Science and Technology*, 33(5):1039–1055.

Mahmood, T., Ricci, F., Venturini, A., and Höpken, W. (2008). Adaptive Recommender Systems for Travel Planning. In *Information and Communication Technologies in Tourism*, pages 1–11, Vienna, Austria.

Mendoza, M. and Torres, N. (2020). Evaluating content novelty in recommender systems. *Journal of Intelligent Information Systems*, 54(2):297–316.

Mimoun, M. S. B., Poncin, I., and Garnier, M. (2017). Virtual Sales Agents: The Reasons of Failure. In *The Customer is NOT Always Right? Marketing Orientationsin a Dynamic Business World*.

Naiseh, M., Jiang, N., Ma, J., and Ali, R. (2020). Explainable Recommendations in Intelligent Systems: Delivery Methods, Modalities and Risks. In *Research Challenges in Information Science*, Lecture Notes in Business Information Processing, Cham. Springer International Publishing.

Nalmpantis, O. and Tjortjis, C. (2017). The 50/50 Recommender: A Method Incorporating Personality into Movie Recommender Systems. In *Engineering Applications of Neural Networks*, pages 498–507. Springer International Publishing.

Narang, K., Song, Y., Schwing, A., and Sundaram, H. (2021). FuseRec: fusing user and item homophily modeling with temporal recommender systems. *Data Mining and Knowledge Discovery*, 35(3):837–862.

Narducci, F., de Gemmis, M., Lops, P., and Semeraro, G. (2018). Improving the User Experience with a Conversational Recommender System. In *AI*IA 2018 – Advances in Artificial Intelligence*, pages 528–538. Springer International Publishing.

Prasad, B. (2005). HYREC: A Hybrid Recommendation System for E-Commerce. In Muñoz-Ávila, H. and Ricci, F., editors, *Case-Based Reasoning Research and Development*, Lecture Notes in Computer Science, pages 408–420, Berlin, Heidelberg. Springer.

Ramnani, R. R., Sengupta, S., Ravilla, T. R., and Patil, S. G. (2018). Smart entertainment - A critiquing based dialog system for eliciting user preferences and making recommendations. In *23rd International Conference on Applications of Natural Language to Information Systems, Paris, France*, volume 10859, pages 456–463.

Sanchez, O. R., Torre, I., He, Y., and Knijnenburg, B. P. (2020). A recommendation approach for user privacy preferences in the fitness domain. *User Modeling and User-Adapted Interaction*, 30(3):513–565.

Sidana, S., Trofimov, M., Horodnytskyi, O., Laclau, C., Maximov, Y., and Amini, M.-R. (2021). User preference and embedding learning with implicit feedback for recommender systems. *Data Mining and Knowledge Discovery*, 35(2):568–592.

Tian, X., Hao, Y., Zhao, P., Wang, D., Liu, Y., and Sheng, V. S. (2021). Considering Interaction Sequence of Historical Items for Conversational Recommender System. In *Database Systems for Advanced Applications*, pages 115–131. Springer International Publishing.

U. S. Food & Drug Administration (2021). Drugs@FDA: FDA-Approved Drugs.

Vajjhala, N. R., Rakshit, S., Oshogbunu, M., and Salisu, S. (2021). Novel User Preference Recommender System Based on Twitter Profile Analysis. In *Soft Computing Techniques and Applications*, Advances in Intelligent Systems and Computing, pages 85–93. Springer.

Ziegler, C.-N. (2005). Semantic Web Recommender Systems. In *Current Trends in Database Technology - EDBT 2004 Workshops*, Lecture Notes in Computer Science, pages 78–89, Berlin, Heidelberg. Springer.