Feature-Driven Interactive Recommendations and Explanations with Collaborative Filtering Approach

Sidra Naveed
Duisburg-Essen University
Germany
sidra.naveed@uni-due.de

Jürgen Ziegler
Duisburg-Essen University
Germany
juergen.ziegler@uni-due.de

ABSTRACT

Recommender systems (RS) based on collaborative filtering (CF) or content-based filtering (CB) have been shown to be effective means to identify items that are potentially of interest to a user, by mostly exploiting user's explicit or implicit feedback on items. Even though, these techniques achieve high accuracy in recommending, they have their own shortcomings- so hybrid solutions combining the two techniques, have emerged to overcome their disadvantages and benefit from their strengths. Another general problem can be seen in the lack of transparency of contemporary RS, where the user preference model and the recommendations that represent that model are neither explained to the current user nor the user can influence the recommendation process except for rating or re-rating (more) items. In this paper, we first enhanced the CF approach by modelling user preferences based on items' features in a complex product domain. The user-feature model is then used as an input to the user-based CF to generate recommendations and explanations. With our proposed approach, we aim to increase transparency and offer richer interaction possibilities in current Recommender Systems- where users are allowed to express their interests in terms of features and interactively manipulate their recommendations through existing user profile and explanations.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Feature-based CF, Interactive recommendations

1 INTRODUCTION

Recommender systems (RS) based on collaborative filtering (CF) and content-based filtering (CB) are widely used techniques [16]. The major difference between CF and CB systems is that CF approach exploits user-item ratings data to generate recommendations, whereas CB systems exploit features of items for recommendations [9].

However, each technique introduces some shortcomings where the CF technique has to deal with a cold-start, scalability and the so called "Gray Sheep"[1] problems [18]. On the other hand, CB systems suffer from over specialization where a user only sees items similar to the ones he or she has already rated in the past, which raises the risk of users being stuck in a "filter bubble" [1].

[1]The term "Gray sheep" refers to a user with unique preferences for which similar users can not be found

ComplexRec 2019, 20 September 2019, Copenhagen, Denmark

Current state-of-the-art approaches are already quite mature and are often applied in a relatively straightforward recommendation scenario by mostly relying on the user ratings of the items. The filtering process of such systems often assumes that the features of an item are equally important for the user. However, in reality, the evaluation of recommendations is a complex scenario especially for high risk-involved domains e.g., digital cameras, where deciding to buy a digital camera is more complex than choosing a song to listen. In such complex domains where personal and financial risk is associated with a product decision, users rely more on item's features to make a decision, where features of an item play an important role in the user's evaluation of an item [4]. However, current CF and CB approaches lack a connection between user ratings and item's features which need to be incorporated especially in the complex recommendation process.

The increasing complexity of recommender systems has also created a demand for more transparent explanations and most of the research on explaining recommendations has mostly focused on a single source of data. Combining the user ratings with item's features in a hybrid manner could more clearly explain the user interests for an item and could be more effective than the explanations that rely only on a single source of data [13].

Additionally, it has been shown that only improving recommendation process in terms of increased algorithmic accuracy does not necessarily lead to appropriate level of user satisfaction [11]. Past research has also shown that in some application domains, users appreciate to be actively involved in the recommendation process and in control of their recommendations [5, 10]. In many, real-world recommender systems e.g., Amazon and Netflix, users have limited or no control to influence their recommendations, to inform the system about its incorrect assumptions, or to specify that preference information has been outdated [2, 7]. These mechanisms are mostly in terms of allowing the users to rate or re-rate single items [14]. Therefore, allowing users to interactively manipulate their recommendations not only lead to higher user satisfaction but also increases the system transparency [8, 12, 19].

To address the above mentioned issues, we implemented a hybrid approach in the complex domain of digital cameras, in which we exploit the user's preferences of features in a collaborative filtering approach that computes the similar users based on the feature preferences rather than the item preferences. We implemented a prototype system, that showcases the possibilities of exploiting the user's preferences of features in terms of 1) incorporating in the complex recommendation process 2) explaining the user's interest of an item 3) interactively manipulating the recommendation process. We contribute to the state of research by addressing the following research question:

RQ 1: How can feature-based information be exploited in collaborative filtering systems for:

a) Preference elicitation of users in cold-start situation

b) Generating item and feature recommendations based on the user profile

c) Generating explanations based on the user profile

d) Manipulating recommendations through existing user profile

2 FEATURE-BASED COLLABORATIVE FILTERING APPROACH

In this section, we describe our implemented feature-based collaborative filtering approach, that enhances the conventional CF approaches which relies mostly on the explicit user's ratings on items.

2.1 Description of the dataset

Recommender systems require a dataset that provides users' interests and preferences. In most of the cases, these preferences are in terms of the set of items and the ratings that users provide to each item. There are many available datasets that can be used to implement different RS algorithms. For our proposed methodology, we used the Amazon ratings dataset, provided by the university of California San Diego (UCSD)[2]. The structure of the dataset is defined by a set of users $U = \{u_1, ..., u_n\}$ and a set of items $I = \{i_1, ..., i_m\}$, where r_i^u are the items assessed, $r_i^u \in D$ (implicit or explicit), by the user u in an expression domain D_u. The rating value $r_i^u \in D$ is defined on the numeric scale from 1 (strongly dislike) to 5 (strongly like).

To implement the feature-based CF approach in the domain of digital cameras, we created a matrix R by only considering the user-items-ratings data of digital cameras. As our approach focuses on exploiting features of an item, it is necessary to complete the information provided by the matrix R with information that describes the product's content in terms of features. This descriptive feature-based information of the cameras has been obtained from a vendor independent test organization (test.de)[3] in Germany and is used to create the item-feature matrix I. This matrix represents a set of items $I = \{i_1, , ..., i_m\}$ described by a set of features $C = \{c_1, ..., c_k\}$, where each item i is described by a vector $V_i = \{v_c^i, c = 1, ..., k\}$.

To associate the user-item-ratings matrix R with item-features matrix I, we created another user-item-features matrix F. For each user u, the matrix F is created and is shown in the Figure 1.

Due to some technical limitations, the features of only 447 digital cameras out of 7659 cameras present in the matrix R, were extracted from test.de website, where 128 cameras out of 447 cameras have ratings present in the matrix R. This reduces our data to 15676 users, 128 items and 16071 ratings, where users who have rated three or more than three items are considered for implementing our prototype system.

2.2 Method

To implement our feature-based collaborative filtering approach, we followed the steps that are explained in the later sections. **1)** Preference elicitation at cold-start: The first step is to create a user's profile based on features instead of items, to use as an input in the feature-based CF approach.

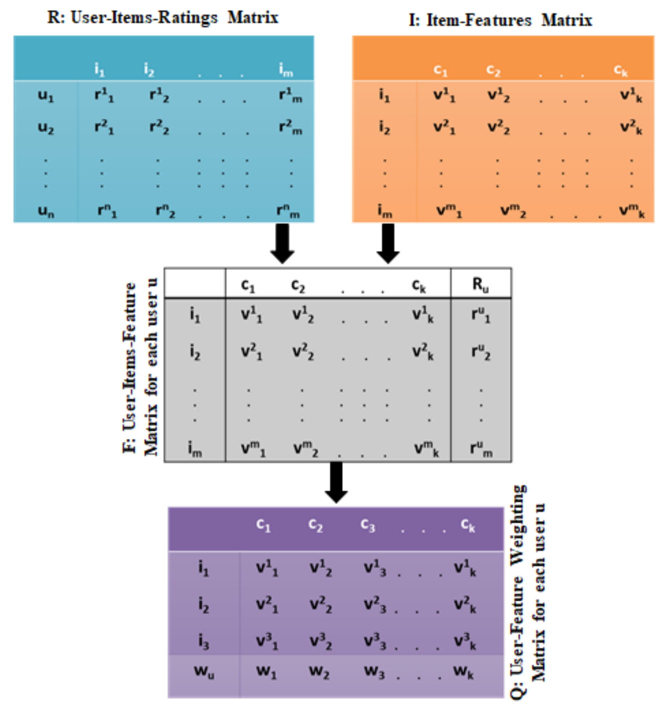

Figure 1: For each user u, the user-feature matrix F is created from matrix R and I, and the user-feature weighting matrix Q is created from the matrix F

2) The user-feature-weighting matrix Q construction: Current CF approaches exploits user's ratings on the items to identify similar users and then predict items from similar users' profiles to recommend it to the active user. The filtering process of such systems, does not takes into account the feature preferences which is subjective and is different for each user. To implement the feature-based CF approach, feature preferences of users are required which could be implicitly derive from the item-ratings [4]. To get these implicit feature-based preferences, we implemented the feature-weighting technique proposed by Barranco et al., [3] (see section 2.2.2) which is composed of three steps: 1) Calculation of inter-user dissimilarity 2) Calculation of Intra-user similarity and, 3) Calculation of feature weights. For performance reason the above-mentioned three steps are computed offline.

3) Computing similar users: The user-feature-weightings matrix Q from the step 2 and the active user's profile are used as an input in our feature-based CF approach to compute similar users based on the feature preferences. To compute similarity between active user's and other users' feature-based profiles, the *Gower's similarity measure* is used [6], which can take into account heterogeneous feature types (nominal and numeric)[4] and the feature-weightings , when computing similarity between objects.

[2]http://jmcauley.ucsd.edu/data/amazon/links.html
[3]https://www.test.de/
[4]For each digital camera, 90-92 features are extracted having heterogeneous data types i.e., nominal, numeric, and binary

4) The item recommendation generation process: Once the similar users are identified, the top rated items from users' profiles are recommended that have all the features preferred by the active user.

5) The feature recommendation generation process: The user-feature-weighting matrix is also used as an input to CF approach to generate feature recommendations based on similar users' feature preferences.

In the following sections, we describe each step in detail.

2.2.1 Preference elicitation at cold-start. In our feature-based CF approach, new users, in contrast to a conventional preference elicitation phase, be asked to select a certain number of features, rate the features in terms of the relevance to their needs using the 5-star ratings, and select the preferred feature-value(s). For numeric features, the user can only select one customized range value and rate the feature, whereas for nominal features, the user can select multiple options using check boxes. This creates a user-feature-ratings profile for the new user. If the user profile already exists and active user updates his/her preferences, the newly selected value for the numeric feature(s) over writes the already present value in the profile, whereas for nominal features, the newly selected value(s) is added to the user profile (if that value does not already exists in the profile).

2.2.2 The user-feature weighting matrix Q construction. [5]

The feature-based preferences of users are missing in conventional item-ratings data sets available for RS, which needs to be derived implicitly from users' ratings profiles. For this purpose, the feature-weighting technique from [3] is applied, which creates a matrix Q, which stores the weightings of features for each user especially when features are multi-valued and heterogeneous in nature. The applied method is further divided into three steps, where each step is explained briefly.

Calculation of inter-user dissimilarity. In this step, the aim is to look for features that may describes the taste and necessities of the user. For this purpose, the entropy H_j is computed for each feature c_j, which takes into account the discriminating aspect of the features i.e., features with more values are more discriminant than features with fewer values. Therefore, the higher entropy value means that the feature is more relevant. To calculate the entropy values for each feature, *Shannon entropy method* is used, which not only give importance to features with more values but also take care of the distribution of these values, where the feature with more uniform distribution gets he higher entropy value [17]. The entropy values are computed using the following formula:

$$H_j = -\sum_{k_j} (f_{k_j}/n) log_2 (f_{k_j}/n) \tag{1}$$

Here, c_k is the feature that takes the set of values k_j, f_{k_j} is the frequency of the feature value v_k in the whole set of items I. The log(0)=0 ensures that the values with frequency 0 does not affect the result. The entropy value H_j is normalized to get the final entropy H_j^* having the values between 0 to 1.

[5]Please refer to [3, 4] for further detail of all the formulas used in section 2.2.2

Calculation of intra-user similarity. In this phase, the dependency coefficient DC_{uk} between ratings provided by the user u on the set of experienced items and the values of a feature c_k on this set of items, is computed. As the features are heterogeneous in nature i.e., numeric and nominal, therefore correlation and contingency measures are used to compute this dependency coefficient for both types of features

- Dependency coefficient for numeric features using Pearson correlation

Pearson correlation is used to measure the dependency between user's ratings on items and the values of the feature k on this set of items, and is computed by the formula:

$$PCC_uj = \frac{\sum_i r_i^u v_{ij}^u - \frac{\sum_i r_i^u \sum_i v_{ij}^u}{n_u}}{\sqrt{\left(\sum_i (r_i^u)^2 - \frac{(\sum_i r_i^u)^2}{n_u}\right)}\sqrt{\left(\sum_i \left(v_{ij}^u\right)^2 - \frac{\left(\sum_i V_{ij}^u\right)^2}{n_u}\right)}} \tag{2}$$

- Dependency coefficient for nominal features using Cramer V coefficient

Cramer V coefficient is used to measure the dependency between user's ratings on items and the values of the nominal feature j on this set of items, and is computed by the formula:

$$VC_{uj} = \sqrt{\frac{\sum_{k_u} \sum_{k_j} \frac{\left(f_{k_u,k_j} - \frac{f_{k_u} f_{k_j}}{n_u}\right)^2}{\frac{f_{k_u} f_{k_j}}{n_u}}}{n_u min(|D_u|, |D_j|)}} \tag{3}$$

Calculation of features weights. The final weight for each feature for each user is computed by taking a product of normalize entropy and dependency coefficient and is given by the formula:

$$w_j^u = DC_{uj}.Hj^* \tag{4}$$

2.2.3 Computing similar users. Once the new user's profile is created or the active user's profile is updated, the next step is to find users that are similar to the active user in terms of the preferred feature-value(s) and the feature-weighting. To compute a similarity, the matrix Q and the profile of active user are used as an input in the user-based collaborative filtering.

As the features of cameras are heterogeneous in nature (nominal and numeric), so separate measures needs to be applied to compute the similarity between two objects, considering the data types of features in to account. For this purpose, Gower's coefficient of similarity [15] is applied, which takes into account nominal and numeric data types when computing similarity between two objects. Additionally, it has an advantage that sparsely populated data matrices are tolerated. For example, it may happen that the active user evaluates a feature c and select a feature value, which is not present in the user-feature-weighting matrix Q of the user u. As a result, no similarity between the active user and the user u could be determined. However, Gower's similarity coefficient solves the problem by assigning a similarity value equal to 0 in the absence of a feature value, without affecting the overall similarity computation.

The Gower's general similarity coefficient measures the similarity between two objects (which in this case are active user and the other user u) based on the variable c, with constant weight w_c, and is computed using the formula:

$$S_{(au,u)} = \frac{\sum_{c=1}^{k} s_{(au,u)c} w_c(x_{(au)c}, x_{(u)c})}{\sum_{c=1}^{k} \delta_{(au,u)c} w_c(x_{(au)c}, x_{(u)c})} \tag{5}$$

Where au denotes the active user, $s_{(au,u)c}$ denotes the contribution provided by the c-th variable between objects au and u and the coefficient $\delta(au,u)c$ determines whether the comparison can be made for the c-th variable between objects au and u. Here $w_c(x_{(au)c}, x_{(u)c})$ indicates that the weight for variable c is a function of variable values $x_{(au)c}$ and $x_{(u)c}$ for objects au and u.

In our approach, we applied the above mentioned method in three steps: 1) Firstly, the similarity coefficient $s_{(au,u)c}$ is computed between the feature values of two objects. , 2) Secondly, the similarity coefficient w_c is computed between the feature weights of two objects, 3) In the third step, the final similarity score $S_{(au,u)}$ between two objects is computed by putting the values of similarity coefficients $s_{(au,u)c}$ and w_c in the equation (5).

Similarity between the feature-values. The first step is to compute the coefficient $s_{(au,u)c_k}$ between the active user and the other user for each value of the feature c_k. As Gower's similarity measure takes into account the type of features when computing similarity, so the method provide separate formulas to compute the similarity coefficient $s_{(au,u)c}$ for nominal and numeric features.

- Computing $s_{(au,u)c}$ for nominal features

The value of $s_{(au,u)c}$ for nominal variable is equal to 1 if $x_{(au)c} = x_{(u)c}$ (objects au and u have the same state for the attribute c) or 0 if $x_{(au)c} \neq x_{(u)c}$ (objects au and u have different state for the variable c). The comparison coefficient $\delta_{(au,u)c}$ is equal to 1 if both objects au and u have observed states for attribute c and zero otherwise.

- Computing $s_{(au,u)c}$ for numeric features

Gower's similarity coefficient $S_{(au,u)c}$ for the numeric features is defined as:

$$S_{(au,u)c} = 1 - \frac{\mid x_{(au)c} - x_{(u)c} \mid}{r_c} \tag{6}$$

where r_c is the range of values for the c-th variable.

In our proposed approach, numeric features have also been considered as nominal features. The reason is the customized range value of a numeric feature that the active user au can select, rather than a distinct value. Because of the range values, the equation (6) can not be applied as it requires the distinct feature value. By considering the numeric features as nominal features, the similarity coefficient $s_{(au,u)c}$ gets the value 1, if the value in the user's u profile lies within the active user's selected range value, otherwise the value is zero.

To compute the coefficient $s_{(au,u)c_k}$ between two users for the feature c_k, the coefficient $s_{(au,u)c_k}$ needs to be computed between the value of the feature c_k for the active user, with all values of the same feature c_k in the user's u profile. The average of the coefficient values for the feature c_k would give the overall contribution of the c-th variable in computing similarity between the active user and the other user.

Similarity between the feature-weights. The next step is to compute the similarity coefficient $w_{(au,u)c_k}$, between the weightings of features in the active user's au and the user's u profile. The weightings computed for users as shown in the section 2.2.2, ranges between [0,1], whereas the active user rate the feature using five-star ratings. To allow the comparison between two users in terms of the feature weightings, the weightings of all features for the other user u is converted to a five-points likert scale using the formula:

$$W^*_{(u,c_j)} = W_{(u,c_k)} * (5-1) + 1 \tag{7}$$

Where, $W_{(u,c_k)}$ defines the old weighting of the feature c_k for the user u. As the feature-weights are the numeric values, so the coefficient $w(au,u)c_k$ is computed by applying the Gower's formula for similarity coefficient of numeric features, using the equation (7).

Computing final similarity score between two objects. The final similarity coefficient $S_{(au,u)}$, which determines the overall similarity between the active user and the other user in terms of feature-values and feature-weightings is computed, by putting the values of coefficients $s_{(au,u)c_k}$ and $w_{(au,u)c_k}$ in equation (5).

2.2.4 *The item recommendation generation process.*

Once the similarity score $s_(au,u)$, is computed between the active user au and all other users, the top 20 users with highest similarity scores are considered as similar users. From the nearest neighbors profiles, the top rated items are selected, and the items that have all the active user's preferred feature-values (again determined by applying Gower's similarity measure), are recommended to the active user.

2.2.5 *The feature recommendation generation process.*

Calculating the prediction scores of features for the active user. In addition to the item recommendations, the features along with the feature values are recommended to the active user, allowing the user to explore diverse features. Once the similar users are identified, the prediction score of the feature c_k, which is not in active user's profile is computed from the similar users' profiles, using the formula:

$$pred_{(au,c_k)} = \bar{r}_{au} + \frac{\sum_{u \in N}(S_{(au,u)}) * (r_{u,c_k} - \bar{r}_u)}{\sum_{u \in N}(S_{(au,u)})} \tag{8}$$

Here, $\bar{r}_a u$ is the active user's average rating for all selected features, N represents the total number of similar users, $r_{(u,c_k)}$ is the user's u rating of the feature c_k, $\overline{r_u}$ is the average rating of the features for the user u, and $S_{(au,u)}$ is the final similarity score of the active user and the other user as computed in section 3.4.2. In the above formula, the feature-ratings of other users are considered to make prediction for a rating of the active user for the feature c_k.

Calculating the feature-values for the active user. Once the prediction scores are computed for each feature based on similar users' profiles, then the top-N features are selected for recommendation. In case of nominal features, the most frequently selected value is used to recommend it to the active user. In case of numeric features, the range of values to recommend are determined by adding and subtracting the standard deviation value from the expected value.

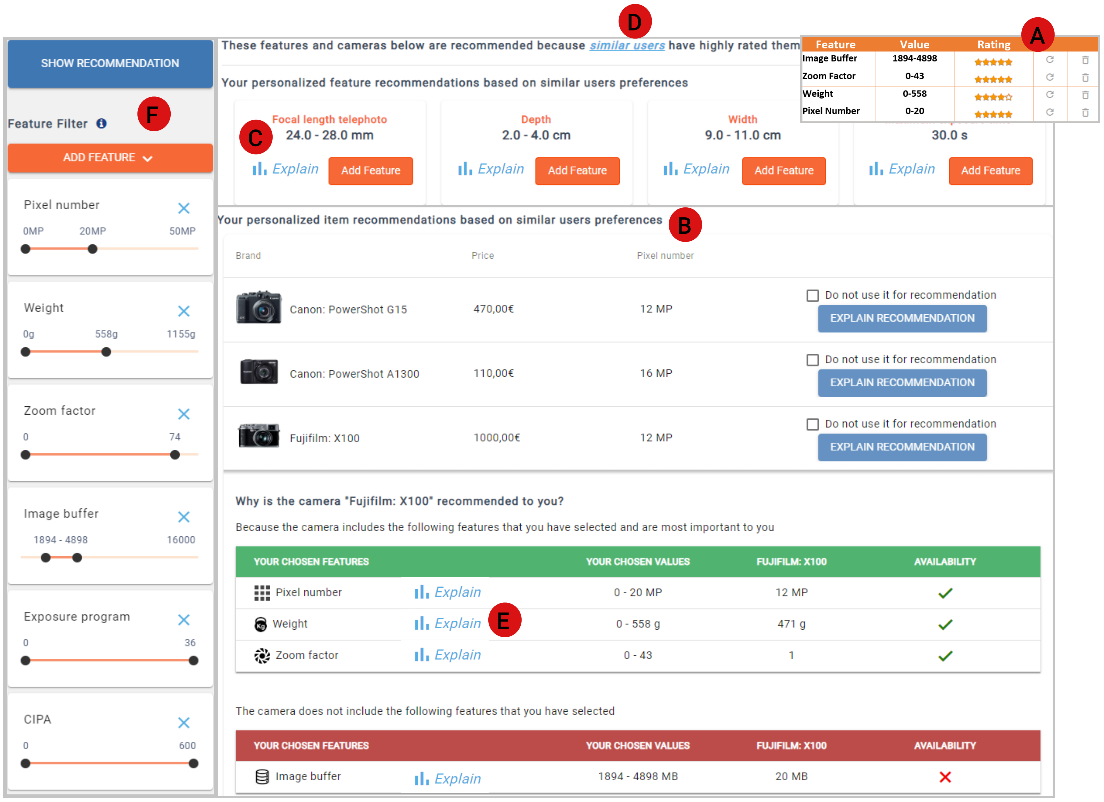

Figure 2: Screenshot of the prototype feature-based CF: (A) representing the user's existing feature-based profile, (B) The item recommendations based on similar users' preferences, (C) The feature recommendations based on similar users' preferences, (D) A link to open the explanation presenting the relevance of similar users with the active user's feature preferences, (E) A link to open the explanation of recommended item based on similar users' and the active user's feature preferences, (F) Filtering area

3 INTERACTION POSSIBILITIES IN FEATURE-BASED CF SYSTEM

In this section, we describe our prototype system which is implemented based on our feature-based CF approach mentioned in section 2, that exploits feature preferences of the active user to compute similar users. The similar users' preferences are then used to not only recommend the items, but also recommend the features to the active user. Our implemented system, provides several interaction possibilities to users allowing them to interactively manipulate their recommendation process through existing preference profile and explanations. The screen shot of our interface is shown in Figure 2, where different components are marked with red alphabetical circles and are explained below.

Eliciting user's preferences in terms of features: As mentioned in section 2.2.1, a new user profile is created based on feature preferences, which is done through filters, where users can select features from the drop down list to indicate their preferences in terms of 5-star rating, and also select preferred feature values (as shown in section (F) in Figure 2). If the user profile already exists, as it can be seen in section (A), any modification in terms of the feature ratings or values, updates the existing profile and thus is reflected immediately in terms of updated recommendations.

The items and features recommendation generation process: Once the new user indicates the preferences in terms of features or the existing user modifies his/her preferences, then this existing profile along with the user-feature-weighting matrix Q (described in section 2.2.2) are used as an input to CF approach to compute similar users based on feature-weightings and feature-values. The section (B) and (C) in Figure 2 shows the item and feature recommendations based on similar users' feature-based profiles.

Explaining item recommendations through feature-based profiles. In our implemented system we provide feature-based explanations which shows that the recommended items generated from similar users' preferences (see section 2.2.3 and 2.2.4) are the clear representation of the active user's feature-based profile. The relevance of similar users with the active user in terms of the active user's preferred features can be seen by clicking on the "similar users" link (shown in section (D) of Figure 2), which further opens the pop-up window shown in Figure 4. This relevance of similar users with the active user can further be explored for each feature of the recommended item, as shown in Figure 3.

Explaining feature recommendations through similar users' profiles. Each recommended feature (see section 2.2.5) is further explained by clicking on the "Explain" link in section (C) of Figure2, showing similar users' preferences for that feature in terms of feature-values, and can be seen in Figure 5.

Manipulating recommendations through existing user profile. In conventional CF approach, the only way provided to users to manipulate their recommendations is by allowing them to (re)-rate item(s). However, our approach not only allows users to improve their preferences by (re)-rating features, but they can also update their long-term profile by adding new features or removing already rated features, as shown in section (A) of Figure 2. This allows the current recommendations which are generated using user's long term preferences, be continuously adapted to user's actual preferences in real time. Furthermore, by clicking on the "add feature" button shown in section C of Figure 2, directly adds the feature along with the feature value into the filtering list, which can be used to update the recommendation process.

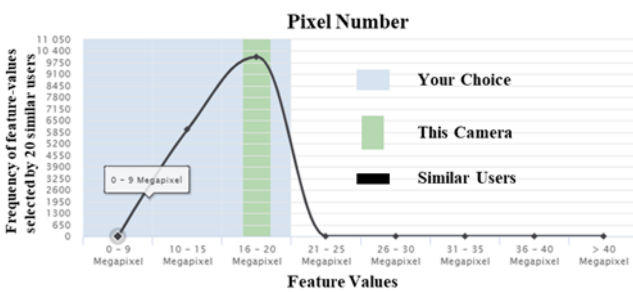

Figure 3: Explanations of the recommended item based on similar users' and the active user's feature preferences

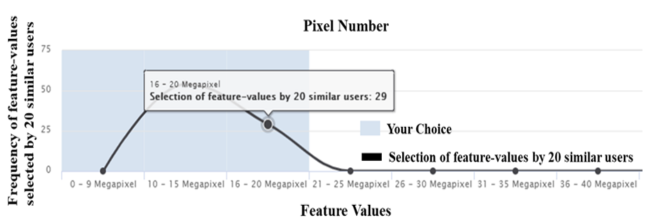

Figure 4: Explanations showing the relevance of similar users with the active user's feature preference

4 CONCLUSION AND FUTURE WORK

In the current work, we extended a conventional CF approach, and exploited feature-based information in CF approach in a hybrid manner to: 1) incorporate the feature-based preferences in complex recommendation process 2) to generate more transparent explanations and, 3) interactively manipulate the recommendation process. Based on the presented feature-based CF approach, we implemented a prototype system that offers several interaction possibilities.

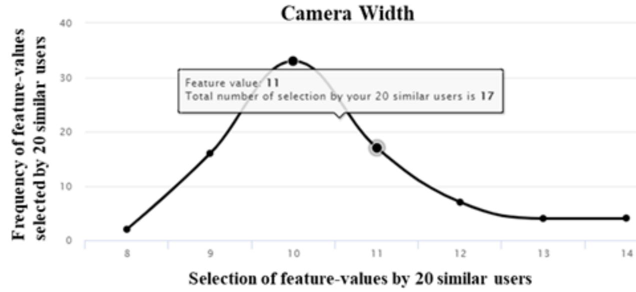

Figure 5: Explanation of the recommended feature based on similar users' feature preferences

In the future, the focus will be on improving the algorithmic accuracy of implemented techniques and explore methods to visually present the explanations in a more transparent manner, which further needs to be evaluated.

REFERENCES

[1] Charu C Aggarwal et al. 2016. *Recommender systems.* Springer.
[2] Xavier Amatriain, Josep M Pujol, Nava Tintarev, and Nuria Oliver. 2009. Rate it again: increasing recommendation accuracy by user re-rating. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 173–180.
[3] Manuel J Barranco and Luis Martínez. 2010. A method for weighting multi-valued features in content-based filtering. In *International conference on industrial, engineering and other applications of applied intelligent systems*. Springer, 409–418.
[4] Jorge Castro, Rosa M Rodriguez, and Manuel J Barranco. 2014. Weighting of features in content-based filtering with entropy and dependence measures. *International journal of computational intelligence systems* 7, 1 (2014), 80–89.
[5] Simon Dooms, Toon De Pessemier, and Luc Martens. 2014. Improving IMDb movie recommendations with interactive settings and filters. In *8th ACM Conference on Recommender Systems (Poster-RecSys 2014)*, Vol. 1247.
[6] John C Gower. 1971. A general coefficient of similarity and some of its properties. *Biometrics* (1971), 857–871.
[7] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets.. In *ICDM*, Vol. 8. Citeseer, 263–272.
[8] Dietmar Jannach, Sidra Naveed, and Michael Jugovac. 2016. User control in recommender systems: Overview and interaction challenges. In *International Conference on Electronic Commerce and Web Technologies*. Springer, 21–33.
[9] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. 2014. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4, 2 (2014), 8.
[10] Bart P Knijnenburg, Svetlin Bostandjiev, John O'Donovan, and Alfred Kobsa. 2012. Inspectability and control in social recommenders. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 43–50.
[11] Joseph A Konstan and John Riedl. 2012. Recommender systems: from algorithms to user experience. *User modeling and user-adapted interaction* 22, 1-2 (2012), 101–123.
[12] Sean M McNee, Shyong K Lam, Joseph A Konstan, and John Riedl. 2003. Interfaces for eliciting new user preferences in recommender systems. In *International Conference on User Modeling*. Springer, 178–187.
[13] Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. 2012. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery* 24, 3 (2012), 555–583.
[14] Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you.* Penguin UK.
[15] János Podani. 1999. Extending Gower's general coefficient of similarity to ordinal characters. *Taxon* (1999), 331–340.
[16] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer, 1–35.
[17] Claude Elwood Shannon. 1948. A mathematical theory of communication. *Bell system technical journal* 27, 3 (1948), 379–423.
[18] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).
[19] Bo Xiao and Izak Benbasat. 2007. E-commerce product recommendation agents: use, characteristics, and impact. *MIS quarterly* 31, 1 (2007), 137–209.