Choice-Based Preference Elicitation for Collaborative Filtering Recommender Systems

Benedikt Loepp
University of Duisburg-Essen
Duisburg, Germany
benedikt.loepp@uni-due.de

Tim Hussein
University of Duisburg-Essen
Duisburg, Germany
tim.hussein@uni-due.de

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

ABSTRACT

We present an approach to interactive recommending that combines the advantages of algorithmic techniques with the benefits of user-controlled, interactive exploration in a novel manner. The method extracts latent factors from a matrix of user rating data as commonly used in Collaborative Filtering, and generates dialogs in which the user iteratively chooses between two sets of sample items. Samples are chosen by the system for low and high values of each latent factor considered. The method positions the user in the latent factor space with few interaction steps, and finally selects items near the user position as recommendations.

In a user study, we compare the system with three alternative approaches including manual search and automatic recommending. The results show significant advantages of our approach over the three competing alternatives in 15 out of 24 possible parameter comparisons, in particular with respect to item fit, interaction effort and user control. The findings corroborate our assumption that the proposed method achieves a good trade-off between automated and interactive functions in recommender systems.

Author Keywords

Recommender Systems; Interactive Recommending; Matrix Factorization; User Interfaces

ACM Classification Keywords

H.3.3. Information storage and retrieval: Information search and retrieval – information filtering.
H.5.2. Information interfaces and presentation (e.g. HCI): User interfaces – evaluation/methodology, graphical user interfaces (GUI), user-centered design.

INTRODUCTION

Recommender systems are often described as systems that have the goal to select from a large set of items—such as products, films or documents—those items that meet a user's interests and preferences best among all alternatives, and to present them to the user in a suitable manner [25]. If successful, recommendations can considerably reduce search effort and facilitate the user's decision process. However, the user's role in current popular recommender techniques such as *Collaborative Filtering (CF)* [29], is very limited. Users can inspect or purchase items once they have been suggested by the system but have no influence on the recommendation process itself, apart from providing implicit or explicit ratings for items which usually only happens in the repeated, longer-term use of the system [18].

Several problems arise from the limited degree of interactivity and user control over the recommendation process. One consequence is a lack of transparency which prevents users from comprehending why certain items are suggested. This is a potential cause of reduced trust in the system [28,30,33]. Also, fully automated approaches often suffer from the widely discussed filter bubble effect [23] which increasingly constrains recommendations to items that are similar to those the user has previously rated positively. This effect makes it more difficult to explore new topics or product categories [15] and to react to situational needs appropriately [7].

Recommender systems research has so far predominantly focused on optimizing the algorithms used for generating recommendations to increase precision [18]. Precision is a measure of how well the suggested items match a user profile based on previously collected data. While precision is an important criterion (and often the only one used) [11,14,22], this narrow view of recommender quality has been criticized for not taking the user's situational needs and goals sufficiently into account [18]. In addition to precision, other metrics have been discussed to measure the quality of a recommendation set, for instance diversity, novelty or serendipity [11,14,31]. Furthermore, most approaches require an existing user profile as input which is often not available (the cold start problem).

While the algorithms currently used for generating recommendations can already be considered quite mature with room only for moderate improvements [18,24], studies have demonstrated that users often desire a more active role in the recommendation process [33], and that interactive control might increase the system's transparency and acceptance. Thus, the development of more interactive recommenders appears to have the potential for increasing the

The definitive Version of Record was published in *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, http://dx.doi.org/10.1145/2556288.2557069.

overall quality of recommendations more significantly than further algorithmic improvements.

Our research goal is, therefore, to increase the level of user control over the recommendation process by combining algorithmic and interactive steps in an *interactive recommending* approach, allowing the user to not only select from a set of recommended items or to criticize presented items, but to influence the recommendation process itself right from the outset. Furthermore, we aim at enabling recommendations in situations where either no user profile exists yet, or the user does not want an existing profile to be applied. In this paper, we introduce an approach that combines the frequently used *Matrix Factorization (MF)* technique [19] with interactive dialogs based on a latent factor model in order to incrementally elicit the user's preferences. This approach combines the potential advantages of automatic methods (accurate recommendations, reduced cognitive load) with the benefits of manual exploration (high flexibility, situational adaptation and high controllability).

The remainder of this paper is organized as follows: The following section discusses approaches for automatic and, especially, for interactive recommendation generation that have been proposed in the past. Next, we introduce our approach, which uses a conventional MF algorithm to characterize items with respect to latent factors derived from a large number of user ratings. We describe how we generate interactive dialogs from the resulting latent feature space that iteratively elicit the users' preferences and provide recommendations. The method developed allows us to minimize the number of steps needed to obtain a sufficiently precise preference profile. After that, we describe a user study we conducted in order to measure the effectiveness and efficiency of our approach. A discussion of the results in the last section concludes the paper.

AUTOMATIC AND INTERACTIVE RECOMMENDER SYSTEMS

Well-established recommenders, such as those used by Amazon [20], Netflix [3] or YouTube [8] have been designed to assist the user in finding interesting content at no or very low additional interaction cost since users can directly select items from the recommendations shown. These approaches are also beneficial in terms of reducing the user's cognitive effort when deciding which item to choose from a large set of mostly unknown choices [24]. However, fully automated recommenders have a number of drawbacks. They are, in particular, not flexible and do not allow the user to control or adapt the recommendation process. Only a few systems allow the user to provide relevance feedback [26] after the recommendations have been shown. While this is a potential way for users to exert influence, the method does not eliminate the filter bubble problem since it only further refines the user's existing interest profile. A further shortcoming of automated recommenders is the lack of transparency for the user who has often no way of determining why certain items are recommended. This may lead to reduced credibility of the suggestions and less trust in the system [28,30,33].

In contrast to fully automated recommenders, interactive search and filter techniques are usually entirely user-controlled and can be regarded as the other end of the spectrum of options for exploring large item spaces. When properly designed, they afford the user the freedom to flexibly explore the item set and also provide a high level of transparency. Yet, they also suffer from several drawbacks. First, they require the user to mentally form a more or less concrete search goal, which is difficult in large and unknown domains. Second, the search and navigation effort is usually significantly higher compared to accepting recommended items. Frequently used interactive methods comprise hierarchical navigation, search input fields and faceted filtering, which, although generally easy to use, may lack the specific filter options that match the user's goal.

The importance of increasing user control over the recommendation process and to improve user experience has recently been pointed out by several authors [18,24,33]. Combining the strengths of recommender algorithms with interactive methods to control the recommendation process appears to be a promising avenue to achieve these goals. From a user perspective, it seems desirable, in particular, to achieve a good trade-off between minimizing the interaction effort needed to identify a relevant item and the level of control over the process. Several approaches have been proposed in the past to reduce the user's initial effort, for instance by active learning [17] or eliciting user feedback through automatically created interview processes [35]. Approaches that combine the training phase and actual recommendations have been used as well, for instance by [34], who use various exploitation-exploration algorithms for this purpose. However, research in this direction has usually been concentrated on algorithmic questions, i.e. finding the optimal subset of items to be rated, rather than improving the user-system-interaction. Also, most approaches are based on persistent user models, although users may not always want this. In the following, we discuss several existing techniques that, in addition, increase the user's control by enhancing recommender systems with interactive features.

Dialog-based recommenders [21], for instance, ask the user a series of questions regarding the search goal, eliciting their preferences to generate appropriate recommendations afterwards. However, prior modeling of item features is required, making such systems costly to develop and only partly flexible.

Critique-based recommenders [6] increase the degree of interactivity by allowing the users to rate the recommendations concerning certain features. This method relies on the assumption that critiquing presented items is often easier for users than forming and expressing their goals up-front. In critique-based recommenders, users can explicitly indicate their preferences, for instance, for cheaper products, a different manufacturer, or items of a different color. Again, the set of features that can be criticized usually depends on previously modeled dimensions. MovieTuner constitutes an exception [32]. It relies on a large set of tags (e.g. “cult film”, “violent”) generated by the users themselves, which allows users to explicitly request movies with, for example, less violence. For this purpose, the most important tags for a particular movie are determined by the system from the entire set of tags and weighted by their specific relevance. While the approach has been shown to be effective, it cannot be applied in all situations, because tags or textual descriptions of the contents must be available.

Recently, a few approaches have been suggested that increase interactivity by combining recommenders with interactive visualization techniques. SmallWorlds [13] is a graph-based interactive visualization for a social recommender embedded in Facebook, simplifying the user interface to state individual preferences. In a user study of this system, the authors found that the recommendation process was more transparent, easier to understand and user satisfaction could be increased. A study of the TasteWeights system came to similar results [4]. TasteWeights is a visual, interactive and hybrid music recommender that allows the user to control the influence of several factors (e.g. preferred artists, social networks). The additional interaction capabilities resulted in a significant gain in perceived recommendation quality. In addition, the visualizations helped the user to better understand the system’s behavior.

The review of related work shows that to date only few and limited approaches exist that blend interactive exploration with algorithm-based recommendation techniques. In addition, with almost all techniques the requirements concerning the underlying data are rather high. Rich predefined datasets, additional content descriptions, or tags are usually necessary to provide the user with more control of the recommendation process. While this may be less of a problem in domains with clearly defined item attributes, it requires that these data are available, and may be restricting in the case of products that depend mostly on the user’s experience (‘experience products’ [27] such as music or movies). To our knowledge, there are currently no effective approaches that increase the user’s control in the described manner without relying on the up-front availability of rich item data.

INTERACTIVE, CHOICE-BASED PREFERENCE ELICITATION FOR COLLABORATIVE FILTERING

In this section, we propose a novel method for combining automated, algorithmic recommender techniques with interactive user control. With this method, we aim to address several objectives: Inspired by systems such as MovieTuner [32], we want to intuitively guide users through an interactive recommendation process, achieving a good trade-off between system support and user control. Users should be able to actively take part in the process without having to know the details of the underlying algorithms. Unlike MovieTuner and many other systems, we aim at providing a solution that does not depend on the existence of metadata such as content descriptions or tags, although we additionally make use of them, if available—but only to improve the items’ presentation. The only prerequisite for our approach is the availability of a standard user-item matrix[1] with numerical ratings, which is typical for Collaborative Filtering [29], the most frequently used method for generating recommendations. This limitation also serves the practical purpose that CF preference data are in many cases more readily available and easier to capture than content-related features. However, our method does not require ratings previously provided by the target user. This also supports both the cold start situation as well as cases where the user does not want to make use of an existing profile.

To make user interaction easy and intuitive, we decided that user preferences should be incrementally elicited by showing system-generated examples of items from which the user selects the preferred ones (“I want something like Shrek or Toy Story”). This approach is partly inspired by Conjoint Analysis [12], a technique often used in marketing research that uses comparisons in order to derive user interests. However, Conjoint Analysis requires a set of predefined feature characteristics and may be error-prone if a user does not know some of the items he or she is supposed to compare. The basic idea behind our approach is, thus, to use latent item features derived from the rating matrix and request preferences for *sets* of similar items instead of single items. To facilitate decision making, we limit the selection to a binary choice, letting the user choose between two sets of sample items in each interaction step. Previous research has shown that users prefer comparing items instead of rating them [16], and ratings also often tend to be inaccurate [1]. Since the number of interaction steps needed should be minimized, we developed a technique based on latent factors to achieve a maximum *information gain* with each choice. With these goals in mind, we propose a method that might be described as “interactive choice-based preference elicitation”. Without loss of generality, we now describe the method in detail by using an interactive movie recommender as a running example.

In our approach, we first extract latent factors from the (typically very large) user-item matrix and assign all items a vector of latent feature values. The latent factors span a vector space with a much lower number of dimensions than the original user-item matrix. While latent factors can be computed with standard SVD (singular value decomposition), current CF recommenders often use Matrix Factorization techniques (usually based on *Alternating Least Squares*

[1] As customary in the recommender area, we define a user-item matrix as a data source that contains numeric ratings, which express a user’s opinion about an item. Without loss of generality, we assume that such a rating is an integer between 1 (“did not like“) and 5 (“liked it very much“).

or *Stochastic Gradient Descent* algorithms [19]), as these are able to handle the typically sparse user-item matrices and lead to very accurate recommendations [19]. This initial step in our approach is identical to the training phase of a typical fully automated MF [19] recommender (in fact, we use an existing MF algorithm[2] for this purpose). Afterwards, the movies are arranged in the *n*-dimensional vector space according to their latent feature values, where *n* is the number of factors[3] taken into account (Figure 1).

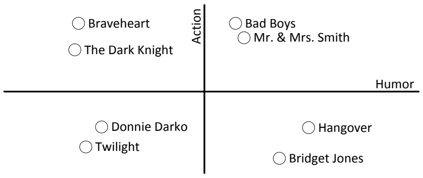


Figure 1. Example of a latent factor model with two factors representing the degree of action and humor in the movies. In contrast to this small example, it is generally not possible to assign meaningful labels to the dimensions.

Although the computed features may relate to real-world concepts and may describe more or less obvious characteristics [19] such as "movies with black humor" or "movies with romantic love stories", it is in general not possible—and not required for our approach—to label them with meaningful concepts due to the method's statistical nature. But, for any given position in the vector space, we are now able to identify a number of movies that match the features represented by this position best by using a standard similarity measure such as the distance between the coordinates.

The standard MF approach for generating recommendations extracts vectors of latent features for both users and items from the rating matrix. It then selects item vectors that are close to the target user's vector based on a given similarity measure and recommends these items [19]. This requires that the target user has already rated a sufficient number of items. In our method, we incrementally position the target user within the vector space in order to generate well-fitting recommendations. We utilize the vector space spanned by the latent factors as a basis for automatically generating interactive dialogs that elicit the user's preferences with respect to these factors.

In our method, a dialog consists of a series of choices between two alternative sets of movies (or a 'don't care' option), where each decision determines the user's position with regard to a single factor. Figure 2 illustrates this approach with an example of two latent factors (each represented by one axis of the vector space). The first factor might, for instance, represent *humor* where factor 2 might describe the degree of *action* in the respective movies.

In each step, one of the two sets shown comprises movies with low values for the currently presented factor *f*, whereas the films in the other set score highly for factor *f*. The values c_{fA} and c_{fB} represent the center points of the intervals containing the currently considered movies (more details about these movie segments in the following subsections).

With each interaction step, a vector *u* representing the user's interests is updated depending on the user's choice. If the user chooses the set with low values S_{fA}, the *f*-th entry in the user vector is set to c_{fA}. In contrast, the entry is set to value c_{fB}, if the user prefers the set of movies with high feature values S_{fB}. If the user prefers not to make a decision, the corresponding dimension is ignored in the following process and the preference vector's *f*-th entry remains undefined. As a final step, we determine those movies whose vectors have the shortest distance from the incrementally positioned user preference vector *u*, i.e. which have very similar latent feature characteristics. These items are finally presented as recommendations.

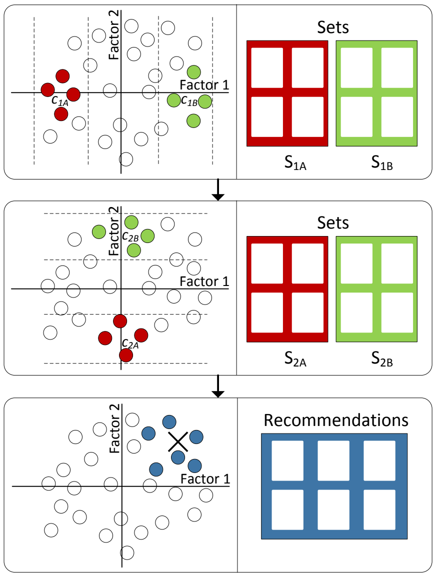


Figure 2. For each factor *f* taken into account, two sets of movies S_{fA} and S_{fB} are presented to the user. One set shows movies with low factor values, the other movies with high factor values. The user selects one of these sets (or indicates that he/she doesn't care). After a defined number of steps, a set of recommendations is computed.

In order to position the user as precisely as possible and to determine an optimal preference vector, we would need to iterate the decision process over each factor. This is, however, not advisable from a usability perspective as MF typically produces up to 100 factors [19]. While precision might increase from step to step, users are likely to get bored or impatient with a larger number of factors, i.e. more dialog steps. Also, distinguishing between the sets of sample movies becomes more and more difficult. The results of a prior study and preliminary experiments with early prototypes suggest that users can well distinguish between the sets, if the number of decisions is limited to approximately 5 factors. A larger number leads to repeated presentation of

[2] *FactorWiseMatrixFactorization* (inspired by [2]) from the MyMediaLite [10] recommender library.

[3] The number of factors (usually 5 to 100) has to be specified before the actual factorization.

some movies and increases the difficulty of understanding the differences between the sets. We thus need to identify the most important (i.e. distinctive) factors and limit the interaction steps accordingly.

With respect to designing the dialogs, the pilot study also showed that the movies representing the different feature characteristics should be chosen with care. Simply selecting those movies that possess minimum or maximum values with regard to the respective factors often did not yield discriminable sets. We will address both of these aspects in the following subsections.

Selecting and Ordering the Factors Used for Positioning the User in the Vector Space

With each interaction step, i.e. each factor taken into account, the user is more precisely positioned in the feature space at the expense of additional interaction costs. Thus, a trade-off has to be established between a "sufficiently good" positioning and the number of interaction steps needed. To achieve this, the selected dimensions should differentiate between the items as much as possible. A standard approach to this problem is to consider the amount of variance explained by a factor. As mentioned earlier, we use a so-called factor wise MF algorithm: Factors are learned one after the other in the order of decreasing percentage of explained variance [2]. Thus, the factors are already ordered by their distinctiveness [9], where the most distinctive factors are more informative and can be assumed to result in choices that are easier for the user than the less distinctive ones. As a consequence, we can limit the number of steps to the most relevant factors. In principle, the number of steps can be determined adaptively depending on the cumulative proportion of explained variance. In our system and study, we chose a fixed number of five interaction steps since it differentiated well the items for the dataset used, and was acceptable to users according to our pretests.

Selection of Appropriate Representatives

A further challenge is to select suitable sample movies which are representative for the low and high values of a factor. Early pilot studies and informal interviews with test users suggested three requirements for the movie sets to be presented to the users:

- *Popularity:* To ensure that the user is able to make a qualified judgment over the two different sets, we restrict the possible representatives to popular movies depending on the total number of ratings a movie had received. This information can easily be gained from the given user-item matrix, from which we selected the 150 most frequently rated movies. Furthermore, we filtered out old movies released before 1960, finally resulting in a list of movies that were generally well-known. Although we used additional data in this last step not contained in the user-item matrix, this does not restrict the general applicability of our approach, since it constitutes a domain-specific optimization step that might be omitted or substituted by other techniques.
- *High diversity between the sets:* The movies should be selected so that both sets are highly distinguishable with regard to the current factor. Since extreme values might distort the decision, we remove the items in the lower and upper fifth percentile for that factor. Afterwards, we partition the remaining items by dividing the item space into 4 segments, each covering an equally-sized interval of factor values. Films in the lower (A) and upper (B) 25% value interval are chosen as candidate representatives (see Figure 3). We use relatively large intervals to ensure that a sufficiently large number of items is available for the selection of appropriate sample movies to be shown.
- *Isolation of the factor:* Additionally, it is important to ensure that the sample movies shown for a factor are as neutral as possible with respect to all other dimensions of the latent factor model. Therefore, after selecting those items that are dissimilar in the current factor, we further filter down the candidate set by selecting movies most similar with respect to all other factors. For this purpose, we construct average vectors x_A and x_B for each segment A and B. The component of x_A or x_B representing the current feature is assigned the average of this feature for all films in A or B, while the other components are filled with the feature averages for all films. These vectors are positioned towards to center of segments A or B while minimizing the distance from the averages of all other features. We can thus assign a weight to all candidate items by determining their item vector's q_i distance from the respective x_j:

  $$\text{weight(i)} = 1 \;/\; \text{dist}(\text{q}_{\text{i,}}\ \text{x}_{\text{j}})$$

  Finally, we select the items with the highest weights as representative examples for the current factor.

Figure 3 illustrates the application of the three criteria described above.

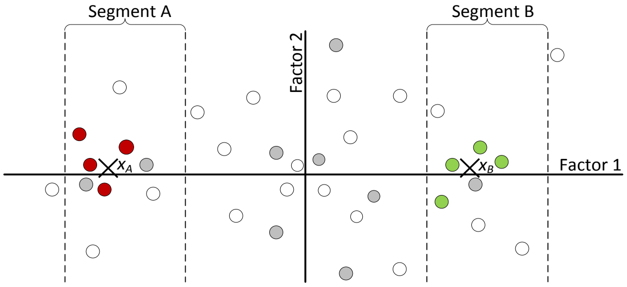


Figure 3. Schematic example for the selection of appropriate representatives. Movies not popular enough (shown in grey) are ignored. The item space is then divided into segments for the currently presented factor (here factor 1), of which only those with movies of low or high factor values are further used (A and B) to obtain movies that are sufficiently different with respect to this feature. Afterwards, the movies near to the average vectors are selected, ensuring that their other characteristics (here only factor 2) are as neutral as possible.

The three criteria have shown to be useful heuristics for selecting item samples that are well distinguishable, specific for the currently considered factors and are likely to be known to the user.

EVALUATION

In order to evaluate the effectiveness of our approach, we developed a prototype recommender system and conducted a user study, comparing our method with three alternatives. Since our interactive approach was developed as an alternative to both manual exploration and (fully automated) CF recommending, we compared it with these two techniques. We assumed that free manual exploration using well-known elements such as text-based search and filtering, would result in items that match the user's interests very well (high effectiveness). On the other hand, we expected manual navigation to be perceived as less efficient because of the typically large number of interaction steps required. For the automatically generated recommendations, we expected the opposite: higher efficiency due to limited interaction, but lower effectiveness in terms of precision of the results. We hypothesized that our approach (in the following referred to as "Interactive Recommendations") represents a good trade-off between these complementary approaches in what can be considered a multi-criteria optimization problem. In addition, we included a simple, popularity-based recommender to obtain a baseline with respect to recommendation fit since these recommendations may be good enough for some users without requiring further interaction or additional data.

Setting

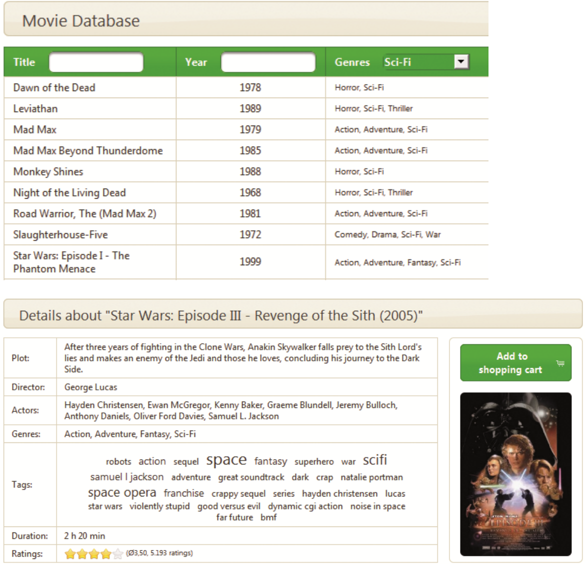
To evaluate our assumptions, we developed a movie web portal offering the four different methods. As background data, we used the MovieLens 10M[4] dataset, which is widely considered as a reference dataset for evaluating recommender systems. Due to the domain independence of CF, we are able to ensure a sufficient degree of ecological validity this way. To provide users with an informative and appealing visual presentation of the movies, we enriched the dataset with plot descriptions, tags, cinema posters and other metadata. For this purpose, we used the HetRec'11 dataset [5] and imported additional data from the Internet Movie Database[5] (IMDb). The web portal offers the following four methods with corresponding user interfaces for exploring the content:

- **Popular films (Pop)** serves a baseline recommender that selects popular movies by a function similar to the one used by the Internet Movie Database (IMDb) to calculate its top 250 movie charts[6]:

$$\text{popularity}(\mathrm{i}) = \frac{k * \bar{r} + c_i * \bar{r_i}}{k + c_i}$$

This function takes into account both the number of ratings c_i for an item i as well as its average rating $\bar{r_i}$. $\bar{r}$ is the mean rating across all items, while k is a constant we set to 100 as the result of early experiments with the dataset. Thus, the average ratings of the items are corrected towards the global mean. Without further interaction, we just presented the top six movies to the user. This selection is not personalized, but results in a list of movies likely to be known such as "Schindler's List", "Pulp Fiction" or "The Matrix".

- **Manual exploration (Man)** allows the users to freely interact with a search and filtering interface (Figure 4). Navigation menus, search forms, tags and hyperlinks could be used to explore the item space. Users were instructed to find six movies they would like to watch, and add them to a list of considered items (to be able to compare the quality of the resulting item sets, each method produced the same number (six movies) of recommended or considered items).



Movie Database

Title	Year	Genres Sci-Fi
Dawn of the Dead	1978	Horror, Sci-Fi
Leviathan	1989	Horror, Sci-Fi, Thriller
Mad Max	1979	Action, Adventure, Sci-Fi
Mad Max Beyond Thunderdome	1985	Action, Adventure, Sci-Fi
Monkey Shines	1988	Horror, Sci-Fi
Night of the Living Dead	1968	Horror, Sci-Fi, Thriller
Road Warrior, The (Mad Max 2)	1981	Action, Adventure, Sci-Fi
Slaughterhouse-Five	1972	Comedy, Drama, Sci-Fi, War
Star Wars: Episode I - The Phantom Menace	1999	Action, Adventure, Fantasy, Sci-Fi

Details about "Star Wars: Episode III - Revenge of the Sith (2005)"

Plot:	After three years of fighting in the Clone Wars, Anakin Skywalker falls prey to the Sith Lord's lies and makes an enemy of the Jedi and those he loves, concluding his journey to the Dark Side.
Director:	George Lucas
Actors:	Hayden Christensen, Ewan McGregor, Kenny Baker, Graeme Blundell, Jeremy Bulloch, Anthony Daniels, Oliver Ford Davies, Samuel L. Jackson
Genres:	Action, Adventure, Fantasy, Sci-Fi
Tags:	robots action sequel space fantasy superhero war scifi samuel l jackson adventure great soundtrack dark crap natalie portman space opera franchise crappy sequel series hayden christensen lucas star wars violently stupid good versus evil dynamic cgi action noise in space far future bmf
Duration:	2 h 20 min
Ratings:	★★★★☆ (Ø3,50, 5.193 ratings)

Figure 4. Two partial screenshots of the manual exploration interface. The upper image shows a list of movies with various filter options. The lower image presents a detail page for a particular movie (with director, actors, genres and tags marked as hyperlinks used for navigating to a list of corresponding movies).

- **Automatic recommendations (Aut)** presents six recommendations generated by an unmodified MF algorithm. We used the *MatrixFactorization* recommender from the MyMediaLite library [10] and initialized it with ten factors[7]. To generate recommendations, initial

[4] http://www.grouplens.org/node/12 (The MovieLens 10M dataset contains about 10 million ratings and 95 000 tags from more than 70 000 users for over 10 000 movies.)

[5] http://www.imdb.com

[6] http://www.imdb.com/chart/top

[7] While this number of factors is relatively low, we gained an RMSE of 0.858 using a 10-fold cross validation on the MovieLens 1M dataset, which is up to standard.

ratings by the current user must be available. At the beginning, users were therefore asked to rate 10 movies out of the 30 most popular items on a 1–5 rating scale. Next, six recommendations were then generated without requiring further interaction.

- **Interactive recommendations (Int)** implements the method described in the previous section, presenting six recommended movies after five decision steps, i.e. the preferences of the users are elicited with respect to the five most important latent factors. Figure 5 shows a sample dialog step where movies that score low and high on a single factors are juxtaposed on the left-hand and right-hand side of the screen. To support the user in making his/her choice, we present additional information for each item or set (not shown in Figure 5). This includes film metadata (movie poster, plot, director, actors, etc.) as well as a tag cloud of terms (e.g. "action" vs. "drama") available in the dataset which is shown below the two movie sets to provide more meaning to the factor currently shown.

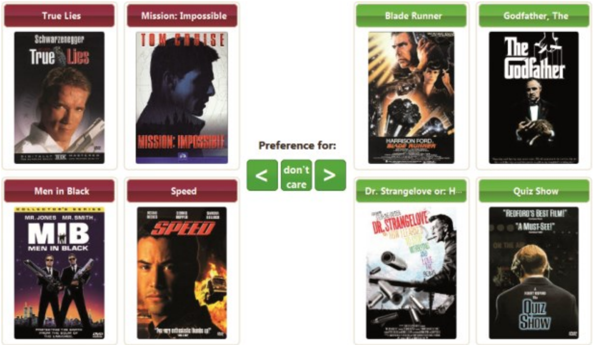


Figure 5. Screenshot showing two movie sets that differ strongly in a single factor. While the left set contains low-brow action movies, the right-hand side displays more serious movies with a rather dark mood.

Hypotheses

To test our assumptions which are based on the goal of achieving an optimal trade-off between different criteria, we formulated the following hypotheses: Our method is superior to at least two of the alternatives with respect to the perceived fit of the resulting items with the user's interests (**H1**), and to the perceived novelty (**H2**) of the recommendations. The methods differ with respect to the user's perceived control over the selection process (**H3**) and with respect to the perceived effort that is required to obtain results (**H4**) —with the interactive method being superior to at least two other methods. Furthermore, the perceived degree of adaptation to the user differs for the four interfaces (**H5**). There are differences in the degree to which users trust the system (**H6**). The interfaces are differently well suited if the user has already formed a search goal to some extent (**H7**). Finally, the methods are also differently well suited if the user does not have a search goal in mind (**H8**). In all of these aspects, we assumed again that our approach is superior to at least two of the alternative conditions.

Method

We recruited 35 participants (24 male, 11 female, average age of 29.54, σ 7.81). The study was conducted over two weeks under controlled conditions under the guidance of a supervisor. The participants used a desktop PC with a 24" LCD-display and a common web browser. Prior to the experiment we collected demographic data and asked participants about their interest in and familiarity with movies.

In the experimental phase, participants used the four methods in a within-subject design. The four different methods Pop, Man, Aut and Int were presented to the participants sequentially in counter-balanced order. Participants were asked to perform one method-specific task per method to obtain a selection of six recommended or considered movies. After each task, participants filled in a questionnaire that was designed to measure the dependent variables corresponding to the hypotheses H1–H8. The questionnaire contained statements with a 7-point bipolar scale ("absolutely not agree" to "totally agree"). In detail, the following statements were presented:

1. The selection fits very well with my movie interests (Fit).
2. The selection contained movies, which I probably would never have found otherwise (Novelty).
3. I felt that I was in control of the selection process at all times (Control).
4. The effort necessary to obtain a selection was acceptable (Effort).
5. I always had the feeling that the system learns my preferences (Adaptation).
6. I trust the system that it takes only my needs into account and not the goals of the system provider (Trust).
7. I would use this method, if I have at least a vague search direction in mind (With direction).
8. I would use this method, if I do not have a vague search direction in mind (Without direction).

Results

Most of the 35 participants reported that they are quite interested in movies. About 85% agreed or totally agreed with a corresponding statement in the questionnaire (giving a rating of a least 5). The participants also stated that they watch about 7.89 (σ 5.88) movies per month, and rated their general knowledge of movies rather high (4.34, σ 1.31).

Table 1 presents the mean values and standard deviations for all eight questions corresponding to hypotheses H1–H8. Thus, strengths and weaknesses of each respective method are outlined. Applying a one-factorial, repeated measures ANOVA (Greenhouse-Geisser adjusted where sphericity was violated) we observed highly significant differences between the four conditions.

		Pop	Man	Aut	Int	F	p
Fit	*m*	4.57	6.17	4.71	5.54	16.984	p=.000
	σ	1.29	1.18	1.51	1.04		
Novelty	*m*	2.91	2.91	4.86	4.80	16.177	p=.000
	σ	1.77	1.84	1.70	1.69		
Control	*m*	1.29	6.31	4.51	5.60	110.488	p=.000
	σ	0.71	1.30	1.72	1.33		
Effort	*m*	6.89	3.49	5.17	6.20	52.319	p=.000
	σ	0.32	1.93	1.60	0.83		
Adaptation	*m*	1.63	1.66	4.94	5.46	96.151	p=.000
	σ	1.06	1.33	1.55	1.22		
Trust	*m*	3.17	5.86	4.69	5.31	27.378	p=.000
	σ	1.65	1.68	1.51	1.23		
With direction	*m*	2.54	5.31	3.63	4.14	31.189	p=.000
	σ	1.70	1.45	1.54	1.33		
Without direction	*m*	4.91	3.34	5.14	5.94	21.713	p=.000
	σ	1.70	2.04	1.38	0.97		

Table 1. Results for the eight questionnaire items (rows) for each method (columns). Standard deviation σ is shown below the mean *m*. All results are highly significant.

As the ANOVA shows significant differences between the four conditions for all questions, we subsequently performed a pairwise comparison between our method (Int) and all other methods, applying a post hoc Bonferroni test. Table 2 shows these additional results of comparing Int with the other conditions with respect to mean values. We used arrows ↑↓ to indicate whether Int scored better or worse than the alternative methods, and whether these differences are significant.

	Pop	Man	Aut
Fit	↑ r=.005*	↓ r=.100	↑ r=.082
Novelty	↑ r=.000*	↑ r=.000*	↓ r=1.000
Control	↑ r=.000*	↓ r=.086	↑ r=.012*
Effort	↓ r=.000*	↑ r=.000*	↑ r=.001*
Adaptation	↑ r=.000*	↑ r=.000*	↑ r=.711
Trust	↑ r=.000*	↓ r=.520	↑ r=.030*
With direction	↑ r=.000*	↓ r=.003*	↑ r=.374
Without direction	↑ r=.009*	↑ r=.000*	↑ r=.017*

Table 2. Results of comparing Int with all other methods. Arrows indicate whether Int received better (↑) or worse (↓) ratings than the alternative methods. Significant differences *r* in the post hoc Bonferroni test are marked with * (only significance at the 5%-level indicated here).

The comparison of Int with the alternative methods shown in Table 2 provides evidence that our approach leads to a good trade-off in the multi-criteria problem at hand. It achieved better results than both automatic methods in terms of fit (**H1**), but only the difference to Pop was significant. Not surprisingly, Man obtained a higher fit rating than Int but this difference was not significant.

The novelty of the items selected with Int was rated significantly better than with Pop or Man, and only slightly (not significant) inferior to Aut (**H2**).

With respect to control (**H3**) our approach was superior to Pop and Aut (significant). As expected, the condition Man achieved a slightly better result here, but the difference was not significant. Conversely, Int scored significantly better than Man (and, also Aut) in terms of effort (**H4**). As no interaction was required, Pop achieved a better result here.

With respect to adaptation (**H5**), Int was superior to all other methods (two out of three comparisons were significant). Regarding trust (**H6**), the results were similar to those of user control (H3).

With a given search direction, the suitability of Int was better than Pop and Aut (**H7**), while Man achieved a significantly better result, as expected. Furthermore, Int scored significantly better than all other methods in cases where users have no clear search direction (**H8**).

Finally, we asked the participants which of the four recommendation or exploration approaches they would like to use more often if they were available. Figure 6 depicts their agreement to the corresponding additional statement in the questionnaire. The results suggest that the participants appreciated the new method, and the increased interactivity and controllability in the recommendation process it offers.

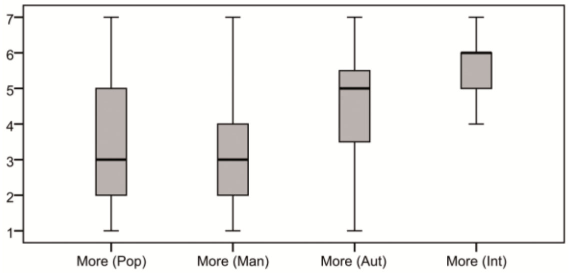


Figure 6. Box plot depicting the participants' stated intention to use the methods.

Discussion

In the user study conducted, our approach leads to significantly better results than the three alternative methods in 15 out of 24 parameter comparisons. As expected, manual exploration showed advantages with respect to fit, perceived control, trust and the usefulness when the search goal is known, although not all differences were significant. In contrast, our approach attained significant better results in all four other aspects, especially in terms of the perceived effort. The advantage of our method over manual exploration in terms of effort must be provided with a caveat, however. To allow for comparing the fit of the resulting item sets, we asked users to also select six items in the manual condition. In a realistic setting, users might stop searching once they found at least one fitting item. Further investigation into performance differences will be needed to account for different task settings.

The automatic condition yielded highest scores in terms of novelty, but our approach was superior in all other aspects. While not all these differences were significant, the interactive approach significantly scored better in a number of other aspects, e.g. trust or suitability for an unclear search direction. However, we want to remark that the slightly negative assessment of the automatic recommender regarding the quality of the recommendations might be influenced by the limited tuning of the algorithm. With more user ratings available ex ante, and perhaps better preference elicitation methods, better results could likely be achieved. But this also applies to the interactive method which could be enhanced by existing user preference data. Regardless of possible weaknesses of the competing methods, our approach leads to recommendations that match the user's preferences very well. Overall, the interactive approach seems to offer a good cost-benefit ratio. It is considered as adaptive, trustworthy and seems particularly useful when the user has not yet formed a search goal.

CONCLUSIONS AND OUTLOOK

In this paper, we present an approach to interactive recommending that combines the advantages of algorithmic techniques with the benefits of user-controlled exploration in a novel manner. We found that the exploitation of latent factors is a promising means to generate interactive, choice-based recommendation dialogs, even when no data about the user are available yet. This allows for producing useful recommendations even in cold-start situations and supports users in their situational needs that may deviate from their long-term interest profile. In comparison to fully automated methods, we achieve a higher level of user-control without sacrificing interaction efficiency. Therefore, the method appears to achieve a good (even if perhaps not yet optimal) trade-off between automated, algorithmic support and interactive exploration of the item space. Showing the user items that score low or high on a limited number of latent factors can be seen as one specific way to sample the item space. While there are various other possibilities to draw samples from this space, as, e.g., with conjoint-analytic techniques, our method aims at maximizing the information gain within few interaction steps and has no requirements in terms of structured item descriptions which in many domains are hard to obtain. Most existing interactive approaches need such additional data up-front. Our core method, in contrast, is novel in this respect and entirely relies on ratings of other users. Thus, comparing it with e.g. MovieTuner would not address the main advantages of our approach. Still, we plan on conducting further comparative studies with other interactive methods in the future. Overall, the choice-based preference elicitation reduces cognitive load and seems particularly useful when users do not yet have a clear search goal or find it difficult to express their needs.

The results of the user study show advantages over three alternative approaches in almost two thirds of all parameter comparisons. In particular, there are clear benefits with respect to item fit, adaptation, effort, and when users do not have a search goal. It is also interesting to see that the method seems to increase trust in the system, even though it uses algorithmic components that are intransparent to the user. One must note a number of limitations of this study, though. First of all, we collected only subjective data based on user ratings which should be complemented by more objective performance data in follow-up studies. Until now, we have focused our evaluation on the user's perception of the interaction process and the resulting recommendation quality with the aim of determining how well our approach performs with respect to an entire set of criteria, instead of comparing absolute execution times. However, we plan to collect such data in future in-depth evaluations and examine efficiency-related aspects in direct comparisons with other interactive approaches. Furthermore, the competing methods might have been better designed or more extensively trained to achieve better results. This, however, applies to our approach as well. The experimental conditions could also be modified in several ways, in particular the setting and tasks could be changed to increase validity. Although our approach can be expected to work well in other domains—as CF-algorithms, which form the basis of our method, are generally regarded as highly domain-independent due to their statistical nature—future work will investigate the approach in other domains such as online shopping. In spite of the limitations mentioned, we believe the current study already provides sufficient evidence that the proposed technique is useful and promising for further research.

In future work, we aim at further optimizing the methods for positioning the user in the item space. There is also room for improving the selection and visualization of sample items to better support the user's decisions, as well as for integrating additional explorative techniques. We also intend to examine the influence of different user characteristics, e.g. familiarity with the product domain or user specific decision strategies.

REFERENCES

1. Amatriain, X., Pujol, J. M., Tintarev, N., and Oliver, N. Rate it again: Increasing recommendation accuracy by user re-rating. In *Proc. RecSys 2009*, ACM (2009), 173–180.
2. Bell, R. M., Koren, Y., and Volinsky, C. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proc. KDD 2007*, ACM (2007), 95–104.
3. Bennett, J., and Lanning, S. The netflix prize. In *Proc. KDD Cup and Workshop 2007*, ACM (2007), 3–6.
4. Bostandjiev, S., O'Donovan, J., and Höllerer, T. Taste-Weights: A visual interactive hybrid recommender system. In *Proc. RecSys 2012*, ACM (2012), 35–42.
5. Cantador, I., Brusilovsky, P., and Kuflik, T. 2nd workshop on information heterogeneity and fusion in recommender systems (HetRec 2011). In *Proc. RecSys 2011*, ACM (2011).
6. Chen, L., and Pu, P. Critiquing-based recommenders: Survey and emerging trends. *User Modeling and User-Adapted Interaction 22*, 1-2 (2012), 125–150.

7. Chi, E. H. Transient user profiling. In *Proc. Workshop on User Profiling* (2004), 521–523.

8. Davidson, J., Liebald, B., Liu, J., Nandy, P., van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., and Sampath, D. The YouTube video recommendation system. In *Proc. RecSys 2010*, ACM (2010), 293–296.

9. Feuerverger, A., He, Y., and Khatri, S. Statistical significance of the netflix challenge. *Statistical Science 27*, 2 (2012), 202–231.

10. Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. MyMediaLite: A free recommender system library. In *Proc. RecSys 2011*, ACM (2011), 305–308.

11. Ge, M., Delgado-Battenfeld, C., and Jannach, D. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proc. RecSys 2010*, ACM (2010), 257–260.

12. Green, P. E., and Srinivasan, V. Conjoint analysis in consumer research: Issues and outlook. *Journal of Consumer Research 5*, 2 (1978), 103–123.

13. Gretarsson, B., O'Donovan, J., Bostandjiev, S., Hall, C., and Höllerer, T. SmallWorlds: Visualizing social recommendations. *Computer Graphics Forum 29*, 3 (2010), 833–842.

14. Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems 22*, 1 (2004), 5–53.

15. Iacobelli, F., Birnbaum, L., and Hammond, K. J. Tell me more, not just more of the same. In *Proc. IUI 2010*, ACM (2010), 81–90.

16. Jones, N., Brun, A., and Boyer, A. Comparison instead of ratings: Towards more stable preferences. In *Proc. WI-IAT 2011*, IEEE (2011), 451–456.

17. Karimi, R., Freudenthaler, C., Nanopoulos, A., and Schmidt-Thieme, L. Exploiting the characteristics of matrix factorization for active learning in recommender systems. In *Proc. RecSys 2012*, ACM (2012), 317–320.

18. Konstan, J. A., and Riedl, J. Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction 22*, 1–2 (2012), 101–123.

19. Koren, Y., Bell, R. M., and Volinsky, C. Matrix factorization techniques for recommender systems. *IEEE Computer 42*, 8 (2009), 30–37.

20. Linden, G., Smith, B., and York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing 7*, 1 (2003), 76–80.

21. Mahmood, T., and Ricci, F. Improving recommender systems with adaptive conversational strategies. In *Proc. HT 2009*, ACM (2009), 73–82.

22. McNee, S., Riedl, J., and Konstan, J. A. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *Ext. Abstracts CHI 2006*, ACM (2006), 1097–1101.

23. Pariser, E. *The Filter Bubble: What the Internet is Hiding From You*. Penguin Press, 2011.

24. Pu, P., Chen, L., and Hu, R. Evaluating recommender systems from the users perspective: Survey of the state of the art. *User Modeling and User-Adapted Interaction 22*, 4-5 (2012), 317–355.

25. Ricci, F., Rokach, L., and Shapira, B. *Recommender Systems Handbook*. Springer, 2010, ch. Introduction to Recommender Systems Handbook, 1–35.

26. Salton, G., and Buckley, C. Improving retrieval performance by relevance feedback. In *Readings in Information Retrieval*. Morgan Kaufmann, 1997, 355–364.

27. Senecal, S., and Nantel, J. The influence of online product recommendations on consumers online choices. *Journal of Retailing 80*, 2 (2004), 159–169.

28. Sinha, R., and Swearingen, K. The role of transparency in recommender systems. In *Ext. Abstracts CHI 2002*, ACM (2002), 830–831.

29. Su, X., and Khoshgoftaar, T. M. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence 2009* (2009), 1–19.

30. Tintarev, N., and Masthoff, J. *Recommender Systems Handbook*. Springer, 2010, ch. Designing and Evaluating Explanations for Recommender Systems, 479–510.

31. Vargas, S., and Castells, P. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proc. RecSys 2011*, ACM (2011), 109–116.

32. Vig, J., Sen, S., and Riedl, J. Navigating the tag genome. In *Proc. IUI 2011*, ACM (2011), 93–102.

33. Xiao, B., and Benbasat, I. E-commerce product recommendation agents: Use, characteristics, and impact. *MIS Quarterly 31*, 1 (2007), 137–209.

34. Zhao, X., Zhang, W., and Wang, J. Interactive collaborative filtering. In *Proc. CIKM 2013*, ACM (2013), 1411–1420.

35. Zhou, K, Yang, S.-H., and Zha, H. Functional matrix factorizations for cold-start recommendation. In *Proc. SIGIR 2011*, ACM (2011), 315–324.