A 3D Item Space Visualization for Presenting and Manipulating User Preferences in Collaborative Filtering

Johannes Kunkel
University of Duisburg-Essen
Duisburg, Germany
johannes.kunkel@uni-due.de

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

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

ABSTRACT
While conventional Recommender Systems perform well in automatically generating personalized suggestions, it is often difficult for users to understand why certain items are recommended and which parts of the item space are covered by the recommendations. Also, the available means to influence the process of generating results are usually very limited. To alleviate these problems, we suggest a 3D map-based visualization of the entire item space in which we position and present sample items along with recommendations. The map is produced by mapping latent factors obtained from Collaborative Filtering data onto a 2D surface through Multidimensional Scaling. Then, areas that contain items relevant with respect to the current user's preferences are shown as elevations on the map, areas of low interest as valleys. In addition to the presentation of his or her preferences, the user may interactively manipulate the underlying profile by raising or lowering parts of the landscape, also at cold-start. Each change may lead to an immediate update of the recommendations. Using a demonstrator, we conducted a user study that, among others, yielded promising results regarding the usefulness of our approach.

Author Keywords
Recommender Systems; Interactive Recommending; Matrix Factorization; User Interfaces; User Profiles; User Experience; 3D Visualizations

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*; I.3.8. Computer Graphics: Applications

INTRODUCTION
Recommender Systems (RS) have become a widely adopted means to tackle the problem of information overload users are often confronted with, for instance, on e-commerce websites, in social networks, on hotel booking portals or in online movie stores [52]. To present users with items that meet their interests, different approaches have emerged. *Collaborative Filtering* (CF), the overall most popular recommendation technique, solely relies on user feedback elicited by asking users explicitly to rate items or by implicitly tracking their interaction with the systems [26, 36]. *Matrix Factorization* (MF) represents the most common model-based CF approach, which generally performs best in terms of objective accuracy while being highly efficient [37]. By statistically analyzing existing rating data, latent factors are inferred which in the following can be used to predict a user's ratings for yet unseen items.

For a long time, RS research has solely been focused on issues related to such algorithms, in particular their accuracy and performance. Only recently, it became more and more accepted that user-oriented aspects such as system transparency or the degree of control users are able to exert over the recommendation process considerably contribute to actual user satisfaction [66, 35, 51, 32]. For instance, users may be reluctant to accept recommendations because they do not understand why certain items are recommended [60], which consequently reduces the system's trustworthiness [66, 51]. The widely used presentation of results in form of ranked lists is not very supportive in this regard, since they usually convey only little information about the recommender's internal rationale [45, 28]. Several approaches exist to increase transparency, e.g. through explanations [24, 63, 60]. However, this typically requires additional content data and is particularly difficult when using model-based CF [37, 13]. Moreover, when presenting just top-n recommendations, users are unable to get an overview of the naturally large item space and cannot adequately assess item coverage, i.e. how shown items relate to remaining non-recommended ones. Becoming aware of alternatives and different, possible diverse areas of potential interest is thus rather difficult [46], and increases the risk of users being trapped in "filter bubbles" [47]. In addition, it often remains unclear how expressed preferences actually correspond to the system's representation of the user, i.e. the user model, and how manipulating the preference profile, e.g. by providing further ratings, affects the results.

From an algorithmic perspective, it becomes increasingly difficult to further improve how recommendations are tailored towards the user's actual needs. By providing a higher degree of control over the recommendation process, interactive RS aim at alleviating this problem in various ways [40, 21]. However, in today's RS, results are mostly adapted automatically based on implicit feedback, e.g. viewing or buying actions. To actively influence recommendations, the user's only means is usually to rate single items, either at cold-start or later in the

© Authors 2017. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *IUI '17 Proceedings of the 22nd International Conference on Intelligent User Interfaces*, http://dx.doi.org/10.1145/3025171.3025189.

process. The implicit way to elicit preferences is again prone to be intransparent while the explicit rating of items requires considerable effort on part of the user before receiving fitting recommendations [19, 58, 11]. In addition, ratings tend to be inaccurate [3] and users are shown to often prefer other means than ratings. For instance, comparing items [42] or stating interests on a rather coarse level by selecting and weighting tags [12] can be of benefit—especially for users entering a system [58]. When no or only little information is available for a new user, conventional CF suffers from the well-known cold-start problem, and thus cannot generate accurate results due to lack of data. This may also be the case when a user does not want a profile representing his or her preferences to be persisted, e.g. due to privacy concerns. Even with an existing profile, it can be difficult to recommend items matching the current user's situation since profiles usually describe long-term interests and do not necessarily need to belong to the same person, e.g. when shared between family members.

In this paper[1], we consequently propose an interactive recommending approach, thereby seeking to answer the following research questions:

RQ1: How can the item space in CF be visualized and sampled in a comprehensible manner?
RQ2: How can areas of preferred items be effectively highlighted within this visualization?
RQ3: How can this visualization be used to allow the user to interactively manipulate his or her preference profile, also in cold-start situations?

First, to visualize the item space, we apply model-based CF due to its proven precision and efficiency. In particular, we use a standard MF algorithm, but map the resulting high-dimensional latent factor model onto a two-dimensional surface in which all items are positioned with respect to their similarities. For this purpose, we use *Multidimensional Scaling* (MDS) [6, 28]. By displaying popular and representative sample items, we are then able to provide the user with a comprehensible presentation of the item space only by means of ordinary rating data, i.e. without requiring any other item-related content. Second, we extend the resulting map to also show the preferences of the current user. To reveal areas of interest, and in particular to highlight the items automatically recommended by the system and how they relate to the typically very large rest of the item space, we additionally exploit the third dimension. Therefore, we use the MF predictions for the current user and all items in order to form a landscape where elevations represent areas with high estimated ratings while valleys indicate lower relevance. Finally, this 3D visualization of item space and user preferences allows us to let the user influence the underlying profile that serves to generate recommendations. The user can alter the landscape by creating or reshaping hills and valleys, and thus establish a preference profile in cold-start situations or manipulate an existing one. All changes may immediately be reflected in the recommendations. Since preferences are expressed with respect to entire item regions rather than individual items, this reduces interaction effort and is independent of knowing and rating particular items, which is especially of value when the search goal is vague or the domain unknown.

[1]This paper is a translated and extended version of our previous work published in German [38]. We now describe the method in more detail, present a more developed version of our demonstration system, and report further results from the user study.

The remainder of this paper is organized as follows: First, we discuss work related to visualizations in RS as well as interactive recommending approaches. Next, we describe our method and a prototype system we implemented to demonstrate our approach. Then, we present a user study we conducted to evaluate our method. Finally, we conclude by discussing results and providing an outlook on future work.

VISUALIZATIONS IN RECOMMENDER SYSTEMS RESEARCH AND INTERACTIVE APPROACHES

Increasing the transparency of RS is known to, among others, improve perceived recommendation quality, leading to higher acceptance and more trust in the systems [66, 51, 60]. However, today's automated recommenders often hinder users to understand *how* a system generates recommendations and *why* it recommends certain items [57, 60]. One popular approach to alleviate this problem is to display textual explanations for recommended items [63, 60]. Thus, depending on recommendation algorithm as well as type and amount of available information, recommendations can be explained in several ways. For instance, one can use item attributes and match them with user preferences [63], albeit this requires availability of content data. Social explanations have been shown to be particularly promising in terms of persuasiveness, but are less informative than other variants [55]. When using CF, a very prominent yet simple example is the one of explaining item-based methods, e.g. used by Amazon ("Customers who bought this item also bought…"). Nevertheless, while there exist many early attempts to explain the output of CF algorithms in general [24], especially for model-based approaches such as MF it is still very hard to improve their transparency through explanations [37]. Exceptions such as [13] usually also require additional content information.

Apart from textual explanations, the range of attempts to increase transparency of RS also includes use of visualizations. Rather simple auxiliary graphics depict, for instance, which criteria selected by a user could be fulfilled [41] or which algorithm was responsible for a recommendation in a hybrid setting [49]. But, also more complex visualizations such as flow charts [27], Venn diagrams [49] or even graph-based representations [62] have already been discussed. On a different level, other approaches visualize the user model in order to improve the system's general transparency and the user's understanding of how his or her preferences are represented within the system. Leveraging *Information Visualization* techniques [30, 23], successful examples comprise focus-and-context lists [61], radial displays [4] or icon-based avatars [5].

Particularly in *Information Retrieval*, a considerable number of methods exist for visualizing large datasets such as document collections [22, 1]. Map visualizations, for example, have been shown to be a promising means for facilitating browsing and searching in large collections. Adopted in RS research, maps may also be useful for visualizing the space of available items as well as the user model [44, 65, 17], possibly making the recommendation process more transparent as well as increasing user engagement when interacting with the RS [17, 15]. Assuming a user's preference profile is represented by some high-dimensional vector, it can be projected onto a 2D representation of the item space as a geographical point where items that receive a high predicted rating appear close to this point [31, 15, 44]. Thus, the relation between how the user is modeled and which items are recommended becomes intuitively understandable [31]. However, this kind of maps highly depends on the particular user: They cannot be generated without sufficient information about that user, e.g. at cold-start, and items are arranged differently for each user. Moreover, these approaches usually require a version of the underlying algorithm where the rating prediction is specially geared to create such maps, e.g. by using Euclidean MF [31, 44]. The map visualization of *TVLand* [17], in contrast, is independent of a particular user and his or her estimated ratings. Here, similarities between items are used to create a global representation of the item space. Nonetheless, areas of interest that include the recommendations can still be highlighted by color, similar to a heat map. Consequently, users can see how their preferences expressed through ratings result in the areas and items the system actually suggests. In addition, users are better able to grasp how recommended items are positioned within the rest of the typically large item space, thus allowing them to keep an overview and to become aware of possible alternatives. In conventional RS, this is often difficult due to their use of lists to present recommendations.

Approaches that visualize only a part of the item space, for instance, the region close to the user's position containing the recommended items (as in [31, 44]), may also be prone to this problem. To mitigate the risk of users thus being stuck in a "filter bubble" [47], some visualizations specifically aim at presenting diverse recommendations [48, 65]. Only few exceptions such as the aforementioned *TVLand* [17] visualize the item space as a whole, and at the same time also indicate areas of potential interest. Overall, effectively supporting users through visualizations in RS is still an under-explored field of research, mostly limited to the purpose of explaining recommendations or supporting item space exploration. In addition, possibilities to interact with the visualizations almost never go beyond the means provided in conventional RS.

In CF, preferences are usually elicited via implicit or explicit feedback [26]. However, providing explicit feedback, typically by rating single items, is a tedious task for users that is often decoupled from the actual recommendation process. At the same time it constitutes a very limited means for expressing actual user needs. Thus, this kind of feedback is rather sparsely available [26]. Users who enter a system for the first time have to rate a certain number of items before a CF algorithm can provide them with proper results [11]. To counter this, efforts have been made to keep the number of items to be rated as small as possible [43], to reward users for every rating provided [16], or to seek for alternatives, e.g. comparing items instead of rating them [42]. Also, algorithmic solutions have been suggested with the goal of asking users to rate only the most informative items, e.g. via active learning [14]. However, expressing initial preferences as well as altering an existing profile is nearly impossible in a controlled and transparent manner when the user's only way to influence the recommendation process is to (re-)rate single items.

Therefore, interactive recommending approaches have been proposed that increase user control over the recommendation process. It has been well established that users are generally more satisfied when they can actively influence their search, although this may come along with higher interaction effort and cognitive load [34]. Besides, it has been shown that integrating RS with more interactivity improves, among others, transparency and perceived recommendation quality, which is more decisive than objective accuracy [66, 35, 51, 32]. Increased interactivity may be realized by using other preference elicitation methods than ratings and by eliciting preferences in an ad-hoc fashion, allowing users to immediately observe how their changes affect the results [18, 64, 42, 12]. A greater extent of control seems also beneficial for exploring large item spaces, especially when the search goal is vague [42], and for adapting recommendations towards situational needs. Further, interactive RS may help to alleviate the cold-start problem, and to support users in circumstances where they do not want a persistent profile to be applied, e.g. due to privacy concerns or because it belongs to a different person [64, 7, 42, 12].

Early examples for interactive RS are dialog-based and critique-based approaches. The latter allow users to criticize recommendations based on predefined item metadata [9]. This avoids the problem that users have to formulate their search goal up-front as it is necessary in dialog-based systems. Developments such as *MovieTuner* [64] build on this principle, but rely solely on user-generated content, in particular tags that can be weighted by the user to change the current result set. Other examples of interactive RS comprise *SmallWorlds* [18], *TasteWeights* [7], *SetFusion* [49] or *MyMovieMixer* [41]. These approaches to provide users with more control over the recommendations use manipulable graphs for influencing the underlying CF algorithms [18], interfaces for weighting the different datasources and algorithms in hybrid settings [7, 49], or faceted filtering blended with automated recommendation methods [41]. They all have shown to improve user engagement and overall satisfaction.

To allow users controlling the recommendation process at a more coarse-grained level than providing ratings for single items, these interactive approaches use, for example, tags [64, 12], automatically selected content attributes [7] or predefined item facets [41]. Preference elicitation thus becomes detached from actual items, which indeed has several advantages, but may also result in difficulties. It requires availability of adequate background data and highly depends on the possibility to categorize items among certain dimensions that actually matter to users. Also, mentally establishing a search goal so that preferences can be expressed with respect to specific item features may be non-trivial for users with little domain knowledge or in the beginning of a search task [25]. Only few approaches such as the one proposed in [65] allow users to define areas of interest directly inside the item space. This, however, seems to be a promising and natural way of expressing preferences without having to articulate them explicitly, and without the need to know and rate particular items.

Although several approaches use some kind of visualization, primarily to disclose the reasons for items to be suggested (e.g. [7, 49, 41]), they are typically independent of the more complex, especially map-based visualizations mentioned before. It thus seems promising to visualize the item space of CF recommenders together with user preferences in an integrated fashion by means of a map. This also opens the possibility of increasing user control, and, in particular, of letting users interactively specify their current interests with respect to entire item regions. In sum, an interactive landscape based on item space and user preferences has the potential of facilitating the establishment and manipulation of user profiles.

3D ITEM SPACE VISUALIZATION TO PRESENT AND MANIPULATE USER PREFERENCES IN CF

We propose a method that visualizes the item space together with user preferences as estimated by a model-based CF algorithm, as well as resulting recommendations. The underlying preference profile used to generate recommendations can be interactively set up by the user in cold-start situations and further manipulated in case such a profile already exists. With respect to the research questions posed at the beginning of this paper, the process which is also described in Figure 1 can be divided into the following main steps:

1. Visualize the entire item space as a 2D map and automatically select item samples to be displayed as representatives for the different regions.
2. Present the current user's preferences so that hills indicate areas of high interest, valleys areas of low interest, resulting in a 3D landscape.
3. Allow users to interactively change the elevation profile, this way manipulating the underlying model used to generate landscape as well as recommendations.

In the following, we will describe these steps in more detail.

RQ1: Visualizing and Sampling the Item Space

In order to visualize the item space, we solely rely on common user feedback as is typically used as background data in CF. By using rating data provided by all users, this step is independent of data availability for the current user. Nevertheless, one issue arising when using ratings to plot such a representation is data sparsity, since users typically rate only a small number of items out of the entire item set. Hence, it may be difficult to adequately calculate similarities between items, which is a prerequisite for many algorithms that map high-dimensional data onto low-dimensional spaces. However, it should be noted that although we use explicit ratings, our approach could in principle also be applied to implicit data, which is usually more dense. In either case, handling the large amount of data could lead to decreased efficiency of mapping algorithms. In addition, semantics inherent in these data may only hardly be exploited, potentially tempering quality of the item positioning, thus hindering users to understand the resulting map.

For these reasons, we introduce an intermediate step before plotting items on a 2D surface. In fact, we use a more abstract representation of items by exploiting their description through latent factors as derived by a standard MF algorithm (Figure 1, 1a), which has already been shown to be successful for "putting recommendations on a map" [17].

When using MF, the user-item-matrix $\mathbf{R} \in \mathbb{R}^{|U| \times |I|}$ that contains the raw rating data for all users $u \in U$ and items $i \in I$, is decomposed into two low-rank matrices, namely $\mathbf{P} \in \mathbb{R}^{|U| \times |f|}$ and $\mathbf{Q} \in \mathbb{R}^{|I| \times |f|}$, where f represents a predefined number of factors[2]. These matrices approximate the original user-item-matrix such that calculating the inner product of a user's factor vector $\vec{p}_u$ of $\mathbf{P}$ and an item factor vector $\vec{q}_i$ of $\mathbf{Q}$ returns the predicted rating $\hat{r}_{ui}$ for user u and item i. Estimating a user's ratings for all items is consequently done as follows:

$$\hat{r}_u = \vec{p}_u \mathbf{Q}^{\mathrm{T}} \tag{1}$$

By relying on a latent factor model, we take advantage of the fact that the factors implicitly convey semantics without requiring explicitly defined content data [37, 53, 13], Therefore, a mapping of the item space can be produced that is likely to be understood by users. Moreover, we circumvent any issues that may arise from sparsity, since MF can handle such matrices very efficiently [37]. Finally, by using a MF algorithm at this stage, we can draw on the derived user factor vector (Figure 1, 1a) also in the next step of the process to generate recommendations for the current user. Thereby, we take advantage of the fact that this widely used method is known for high recommendation quality [37, 36].

Next, we map the still high-dimensional item data onto a low-dimensional Euclidean space by using MDS [6, 28]. In order to visualize such data, different methods have been proposed [29, 28]. Typically, they rely on content information, so that the decision for a certain method depends on the item features. Geometric projections and scatter plots have been used very often for this purpose [56]. But, they can also be usefully applied when the dimensions are constructed by automated dimensionality reduction [10]. This is usually the case for datasets used by RS [2]. Thus, although other methods might be used, we chose MDS to calculate two-dimensional coordinates for all items. Using these coordinates, the resulting map visualization positions items based on their similarities (Figure 1, 1b). MDS ensures distances between any two items to be small if they are similar to each other, and large otherwise. We calculate the similarities used as input for the MDS algorithm by means of the Euclidean distance between item factor vectors $\vec{q}_i$, which seems reasonable since it naturally fits the positioning approach of MDS. As shown with the maps generated in [17], we assume that by relying on latent factor representations, it will adequately be reflected how items actually relate to each other. Thus, users should be able to perceive items close to each other as actually similar.

[2]Note that by setting $f = 2$ the item-factor-matrix could indeed be directly represented as a map. However, this would later result in reduced recommendation quality [37].

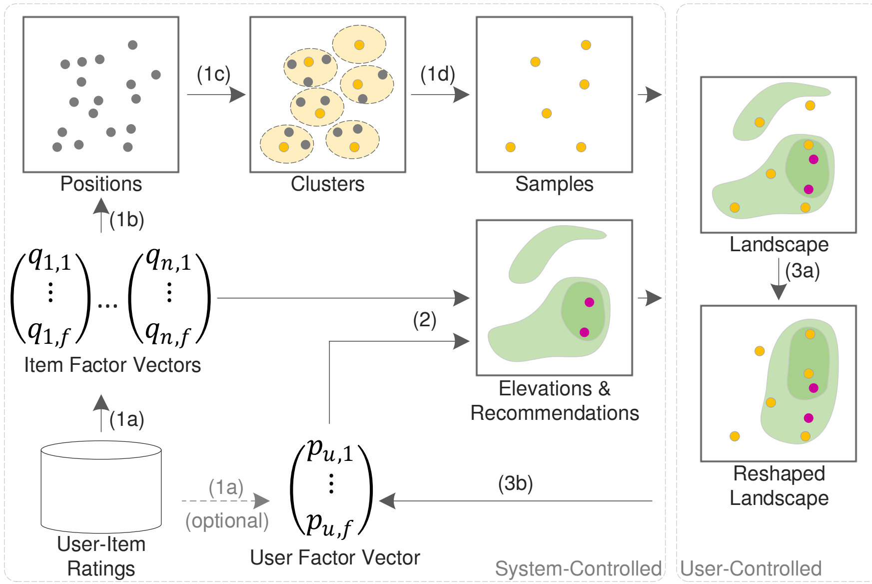

Figure 1. To generate the 3D item space visualization for presenting and manipulating preferences in CF, we start by using Matrix Factorization to obtain latent factors for users and items (1a). Item factor vectors then serve to determine the item positions (gray) using Multidimensional Scaling (1b). By applying k-means clustering (1c), popular representative items are chosen as samples (yellow) to be displayed on the resulting map (1d). In addition, item factor vectors are used to calculate predictions for the current user by taking his or her user factor vector into account (2). This results in elevations representing the user's preferences as well as in the actual recommendations (magenta). Note that the user factor vector is optional, so that this step is left out at cold-start where the user is instead presented with a flat surface. In any case, the user is then able to influence the recommendation process by reshaping the landscape, i.e. creating hills and valleys (3a), which finally leads to recalculation of the user factor vector (3b).

Now, in principle, items could already be plotted onto a 2D surface. However, due to the sheer mass of items, this is not a practical solution and would overwhelm users instead of providing an intuitive overview. Instead of showing popular items and additionally labeling certain areas of the map as in [17], we aim at generating an understanding for the item space based only on items themselves. Therefore, we select items that are representative for different regions of the quadratic map, and present them to the user. To perform this sampling, we use a *k-means* clustering [20] since we are in a Euclidean space (Figure 1, 1c). This allows us to control the trade-off between representativeness and number of items. Then, to determine a representative item for each cluster, we consider the five items that are closest to a cluster center and finally choose the most popular one as a sample, i.e. the item with highest number of ratings (Figure 1, 1d). Chosen items serve as representatives of the respective clusters and are at the same time likely to be known to many users. Based on the map dimensions and these sample items, we render an initial map.

RQ2: Presenting Preferences and Recommendations

In addition to the item space samples in the initial map, we also present the user's preferences and show recommended items in the context of the overall item space. Although the only items shown initially are the samples representing the different regions of the item space, in fact, all items have been assigned a certain position on the map. This allows us to exploit the third dimension by showing a landscape where the elevation indicates the system's predicted preferences among all items in respect to the current user. We therefore use the ratings predicted as usual by MF: Areas containing items with high predicted ratings are visualized as hills, those with low ratings as valleys. Since recommendations lie in areas where the system has predicted items to be of high interest, i.e. on hills, we assume the user will thus better understand how the system models his or her interests, and how this relates to actually expressed preferences as well as recommended items.

If the current user's preferences were previously elicited, e.g. by ratings, a latent factor vector for that user is already available. This vector $\vec{p}_u$ derived in the MF offline learning phase in the previous step (Figure 1, 1a) may now be used to calculate predictions as shown in (1) online. The resulting predictions $\hat{r}_u$ are used directly to select top-n items with highest scores as recommendations, but, as outlined above, also for setting up the elevation profile, and thus the 3D landscape. For this purpose, we linearly map the prediction for every item onto a height value, and consequently set the surface elevation at the item's respective position to this value. Then, to present the user with a visualization that actually resembles a landscape, the elevation of spaces between items is set to a level similar to adjacent items (otherwise, only spikes would appear at every item position). Therefore, we transfer height values of the items to their surrounding area where no items exist in a step-wise manner, decreasing with each step. Afterwards, we apply a Gaussian smoothing function and finally re-adjust the elevation at the actual item positions.

In case rating data for the current user are unavailable or the user does not want to apply an existing profile, i.e. we cannot use the user factor vector, the elevation profile is set to a neutral level. The visualization then shows a flat map surface and samples, both generated independently of the current user.

RQ3: Interactively Manipulating a Preference Profile

Regardless of whether the landscape already represents user preferences or just shows a flat surface when no user profile is available, the user can now interactively influence the underlying model, and consequently the recommendations. By shaping the landscape, i.e. raising or lowering the surface, the user is able to interactively express preferences for entire regions of the item space (Figure 1, 3a). The subsequent recalculation of predictions happens online—either continuously with each interaction, or when explicitly requested by the user, thus avoiding constant, possibly confusing updates of the visualization. In any case, we interpret changes to the landscape as user adjustments of the estimated ratings for the items that have led to the elevation of the respective areas. Note that this is independent of which items are actually shown, but instead takes all items in an area into account. The elevation values changed by the user are used to replace the rating predictions $\hat{r}_u$ calculated previously by new preference values, resulting in the vector $\vec{x}_u$. Based on this, we now set up a new user factor vector or recalculate the existing one by reformulating (1):

$$\vec{p}_u = \mathbf{Q}^+ \vec{x}_u, \tag{2}$$

where we use the pseudoinverse $\mathbf{Q}^+$ to approximate a solution via *Singular Value Decomposition* (since $\mathbf{Q}$ is non-quadratic). The updated user factor vector $\vec{p}_u$ is then fed back into the recommendations process (Figure 1, 3b), where it can be used to again predict ratings, leading to new recommendations as well as a new, adapted elevation profile. Thus, the user can immediately observe how the actions performed affect the recommender's results and the underlying preference profile.

DEMONSTRATOR

In this section, we present a demonstrator for our interactive recommending approach based on the 3D item space visualization described above. We implemented the demonstrator as a web-based application positioned within the movie domain. In addition to demonstration purposes, we also aimed at conducting user experiments with this demonstration system. In the following, we expand on its interaction concept and explain the implementation in more detail.

Interaction Concept

The user interface (Figure 2) is basically divided into four main parts: Working area (A), an area showing recommended items (B), detail area with information on the currently selected movie (C), and a palette of available interaction tools (D).

Within the working area, the visualization generated according to the steps described in the previous section is shown. In addition to the quadratic map surface representing the item space, the sample items, and the hills and valleys indicating the user's preferences, we color the surface to resemble a topographical map. Therefore, we use a function that assigns colors to particular levels of elevation while ensuring smooth transitions between them. This way, we aim at further facilitating the user's perception of the landscape and how it reflects the varying interests. Items are depicted with the help of movie posters directly on the map. Recommended items are additionally highlighted by means of a magenta-colored margin. Recommendations are also shown in the area at the bottom of the screen in form of a more conventional list. When the user hovers over an item on the map or in the recommendation list, the detail area is immediately updated and reveals further information on the respective movie (e.g. title, director, plot description and tags). Note that this content-related data is only used to provide users with additional information, and is not involved in the process of creating the visualization or generating recommendations. Finally, there is a palette showing several tools that may be used to perform interactions within the working area. Each tool has two functionalities that correspond to the left and right mouse button, respectively:

1. *Raise/Dig:* This tool can be used to shape the landscape, i.e. to create hills (left-click) and valleys (right-click) within the quadratic boundaries of the map. If selected, the mouse cursor shows a shovel icon and a small round white area surrounding the cursor indicates where the surface will be altered when clicking[3] (see also Figure 3). The highest or lowest possible elevation is thereby restricted through the linear mapping of predictions onto height values.
2. *Rotate/Pan:* As known from many 3D applications, this tool allows the user to rotate the entire perspective or to pan through the landscape.
3. *Show/Hide:* Inspired by [59], this tool helps to explore the item space in more detail. In case the user wants to see more than the initially shown samples, he or she can bring up additional items by left-clicking on the map (see also Figure 4). Then, the most popular of the five items closest to the cursor gets added. Right-clicking on an item already shown in turn removes this item from the map, which is particularly useful in case the map gets too crowded.

Independent of the tool currently selected, the user can always zoom in and out by using the mouse wheel.

Implementation Details

For implementing the process described, we first use the *Stochastic Gradient Descent* algorithm[4] from the *Apache Mahout*[5] library. This well-proven implementation of a standard MF algorithm allows us to derive the latent factor model used for calculating item similarities and rating predictions with performance up to standard (*RMSE* of 0.80 using 10-fold cross validation). As background data, we utilize the *MovieLens 20M Dataset*[6] containing about 20 million ratings from 137 000 users given to 27 000 movies. In principle, our approach may also be applied to other domains such as books, music, or any other type of commercial goods, in particular because CF, which is the underlying basis for our approach, is generally regarded as domain-independent. However, the MovieLens datasets are well-established within RS research, and, from our point of view, an appropriate means to show that our approach works as expected for *experience products*.

[3]Depending on the demonstrator's configuration, changes to the landscape are fed back into the recommendation process either continuously triggered by every mouse click, or only as soon as the user feels confident with the manipulations and uses the "Apply Changes"-button right underneath the palette (Figure 2, D). More details on how this is done can be found in the previous section.

[4]*ParallelSGDFactorizer* (8 factors, 16 iterations, $\lambda = 0.001$).

[5]https://mahout.apache.org/

[6]http://grouplens.org/datasets/movielens/20m/

Figure 2. Screenshot of our demonstrator: Working area (A) visualizing the item space as a quadratic map that includes movie posters depicting the automatically chosen sample items and represents the user's preferences by the surface elevation; recommended items (B), which are also shown inside the landscape as posters highlighted by a magenta-colored margin; detail information on the currently selected movie (C); and a palette (D) of available interaction tools (in this example, the *Raise/Dig*-tool is selected, which appears at the position of the cursor in the lower middle part of the screen).

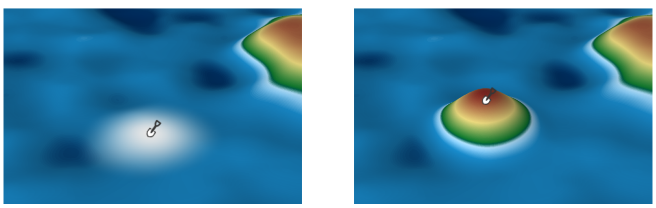

Figure 3. Using the *Raise/Dig*-tool, the user is able to shape the landscape, here by expressing his or her interest through forming a hill.

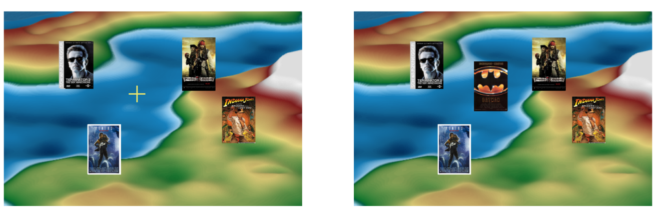

Figure 4. With the *Show/Hide*-tool, further items can requested to be shown in addition to the initially presented samples (inspired by [59]).

This way, we aim at ensuring a sufficient degree of ecological validity. Although not necessary for our 3D item space visualization in general, we additionally enrich the dataset by importing content-related information as well as movie posters from the *TMDb* website[7] in order to provide users with an appealing and informative presentation of the actual items.

Next, based on item factor vectors derived in the MF learning phase, we calculate item similarities which go into a MDS algorithm, resulting in the mapping used to arrange the items on the surface. Therefore, we rely on an implementation by the *Algorithmics Group*[8]. For clustering the items in order to determine representative samples, we use a *k-means* algorithm we implemented ourselves with $k = 30$. Early qualitative experiments suggested this number of initial samples to sufficiently represent the item space while not overwhelming users visually (Figure 2 is print-optimized and shows less samples). Finally, for visualizing the 3D landscape in our web-based application, we use the Javascript 3D library *three.js*[9].

[7] https://www.themoviedb.org/

EMPIRICAL USER STUDY

To evaluate our approach against the research questions, we conducted an empirical user study. We were particularly interested in examining the item space representation and the item sampling, the presentation of user preferences in form of a landscape, the interactive tools for shaping the surface, as well as the effect of these methods at cold-start and with an existing user profile. To assess the effectiveness of our approach, we constructed tasks that focus on these different aspects. We measured the user's perception of different system quality factors, especially with respect to subjective recommendation quality and perceived transparency as well as overall satisfaction and user experience.

Method

Participants and materials: We evaluated our approach using the demonstration system described in the previous section[10].

[8] http://algo.uni-konstanz.de/software/mdsj/

[9] https://threejs.org/

[10] The version of our demonstrator used in the study was slightly different than the one presented in this paper: The interface elements as well as the coloring of the landscape were more simple, and we used an earlier edition of the MovieLens dataset (the 10M version, see http://grouplens.org/datasets/movielens/10m/).

We recruited 32 (10 female) participants with age ranging from 18 to 34 ($M = 24.22$, $SD = 3.61$). The majority had a high school (62.5 %) or a university degree (34.4 %). Participants were asked to use the demonstrator under controlled conditions in a lab-based setting. They used a desktop PC with 24″ LCD (1920 × 1200 px resolution) and a common web browser to interact with the system and to fill in a questionnaire. The interface was in English, but participants (all non-native English-speakers) were explicitly allowed to ask the moderator for translations.

Tasks: The study was structured into three tasks, which were presented to each participant in the same order:

1. *Introductory search and exploration:* The first task can be seen as an introductory task focused on general exploration, orientation in the item space, and familiarizing with the interaction possibilities. In consecutive subtasks, participants were asked to explore the map in order to find three movies fulfilling following criteria: 1a) popular movies (more than 30 000 ratings), 1b) movies suitable for children, and 1c) movies directed by Quentin Tarantino. For each subtask, three minutes were given. Interaction was restricted to exploration, i.e. manipulating the landscape was not possible. No elevations and recommendations were present.
2. *Establishing a profile at cold-start:* In order to evaluate our system in cold-start situations, in this task, participants were asked to express their preferences on a flat surface, i.e. no profile was initially visualized. Starting from the flat surface, participants had to use the available tools to shape the landscape. Participants finished interaction at their own discretion, whereupon a new profile, and thus a user factor vector, was created. Resulting recommendations and the new landscape were presented afterwards to the user.
3. *Manipulating an existing profile:* This task addressed the situation where a user wants to manipulate an existing profile according to his or her current preferences. At the beginning of this task, the elevation profile was set up according to the preferences of an existing user[11]. Participants then had to alter the resulting landscape towards their own preferences. Recommendations and landscape, i.e. the elevations on the map, were updated continuously.

Questionnaires and log data: In order to assess the participants' subjective perception, we used a questionnaire that was primarily composed of different existing constructs[12]. At the beginning of each session, we elicited demographics and domain knowledge (regarding movies and 3D applications). Then, subsequent to task 2, we assessed perceived recommendation quality [33], transparency [50], interaction effort [33] and interaction adequacy [50]. We complemented these existing constructs with a few questionnaire items generated by ourselves, primarily regarding aspects very specific to our approach (e.g. comprehensibility of the landscape and the positions of recommended items inside, perceived controllability of the recommendation process). Next, following task 3, we used the same constructs again, but also assessed perceived control [50] and used self-generated items concerning manipulation of existing profiles. Finally, at the end of each session, we asked participants some questions regarding their general impression of the system. For this, we used constructs such as system effectiveness [33] and perceived usefulness [50], as well as some additional self-generated items. Across tasks, this resulted in about 75 items. In addition, to measure usability, user experience and engagement with the system, we applied the *System Usability Scale* (SUS) [8], *User Experience Questionnaire* (UEQ) [39] and subscales of *Intrinsic Motivation Inventory* (IMI) [54]. All items were assessed on a positive 5-point Likert scale, except the ones from UEQ (7-point bipolar scale) and IMI (positive 7-point Likert scale)[13].

In each session, we also logged interaction behavior, i.e. actions such as selecting tools, shaping the landscape (and how long this took), or showing/hiding items. In addition, we measured task times and, especially for task 1, recorded whether participants were able to accomplish the respective task.

Results

Overall, participants were very satisfied with the system ($M = 3.94$, $SD = 0.76$) and enjoyed using it ($M^{\dagger} = 5.46$, $SD = 0.97$). They perceived the recommender as effective ($M = 3.72$, $SD = 0.74$) and useful ($M = 3.75$, $SD = 0.76$). Table 2 (General results) shows the results for some selected questionnaire items from these general constructs, that particularly emphasize the overall quality of recommendations and the user's enjoyment when using the demonstrator.

Table 1 illustrates results with respect to the general constructs perceived recommendation quality, transparency, interaction effort and interaction adequacy, which we assessed after task 2 and 3, respectively.

	Task 2		Task 3		
	M	*SD*	*M*	*SD*	*d*
Perceived rec. quality	3.57	0.89	3.89	0.60	.42
Transparency	3.91	1.09	3.63	1.07	.26
Interaction effort*	3.75	0.76	3.21	0.93	.64
Interaction adequacy	3.47	0.88	3.61	0.90	.16

Table 1. Differences between task 2 and 3 with respect to perception of recommendations and the interaction (* marks the only construct yielding a significant difference, *d* represents Cohen's effect size value).

In the following, we address our three research questions by expanding on these general constructs and, in particular, by presenting further specific results.

RQ1: Visualizing and sampling the item space: Participants predominantly agreed with the statement that the item positioning on the map was comprehensible and found the landscape helpful for obtaining an overview of the entire item space. Consequently, this facilitated their awareness of possible choice options. Table 2 (RQ1) summarizes the descriptive statistics.

When participants were asked to explore the item space in order to find movies fulfilling different criteria in the subtasks

[11] We carefully selected three existing user profiles from the underlying MovieLens dataset, all very different to each other. Out of these profiles, one was randomly chosen for each participant. Then, in the following, we used the corresponding user factor vector.

[12] We translated questionnaire items to present them in German language, sometimes with slightly adapted formulations.

[13] Mean values of such items are in the following indicated as $M^{\dagger}$.

of task 1, all of them were able to find three movies suitable for children within the given time limit (1b). Popular movies could still be successfully found by 88 % (1a), while only 56 % found three movies directed by Quentin Tarantino (1c). This is also reflected in the time participants needed to accomplish the subtasks: Using a one-factorial RM-ANOVA, we found significant differences, $F(2,62) = 32.801$, $p = .000$. Post hoc comparisons using Bonferroni correction show no difference between task 1a ($M = 1.46$ min, $SD = 0.88$) and task 1b ($M = 1.25$ min, $SD = 0.60$). However, the subtask of finding movies directed by Tarantino took significantly more time ($M = 2.70$ min, $SD = 0.88$) than the two others ($p < .01$).

When we asked participants whether they would use the system in different search situations, we again found significant differences, $F(1.38, 43.01) = 21.010$, $p = .000$. Participants rated the system as very useful for situations where they would have *no* ($M = 3.91$, $SD = 1.25$) or only a *vague* search goal in mind ($M = 4.06$, $SD = 0.98$). Here, post hoc comparisons denote no significance difference. In contrast, participants stated that they would use the system significantly less likely ($p < .01$) in situations with a *concrete* search goal, i.e. for known-item search ($M = 2.44$, $SD = 1.48$).

RQ2: Presenting preferences and recommendations: As already presented in Table 1, perceived recommendation quality and transparency were assessed very positively (no significant differences between task 2 and 3). In addition, the items more specific to our approach reported in Table 2 (RQ2) confirm that the landscape helped participants to understand how their preference profile was represented within the system and that they understood why items had been recommended.

To compare the two tasks, we also assessed the comprehensibility of the generated landscape, i.e. the elevations on the map representing the estimated preferences. We found a significant difference ($t(31) = 2.37$, $p < .05$). In task 2, were participants started from a flat surface, they stated that they understood why the landscape was finally generated the way it was ($M = 3.94$, $SD = 0.91$). When manipulating a profile from another person in task 3, the comprehensibility was rated lower ($M = 3.41$, $SD = 1.04$). Cohen's d, however, suggests only a moderate effect size ($d = .54$).

RQ3: Interactively manipulating a preference profile: The general construct assessing perceived control over the system yielded satisfying results ($M = 3.54$, $SD = 0.94$). When looking at specific questionnaire items regarding the quality of the interaction possibilities provided to express preferences (Table 2, RQ3), scores were even better: Participants felt to be able to tell the system what they like/dislike, i.e. in our case to create hills and valleys, in cold-start situations (assessed after task 2), and to modify an existing preference profile (assessed after task 3). Overall, participants felt in control over the recommendation process by manipulating the landscape.

With respect to perceived interaction effort, Table 1 shows the overall positive results for our system. However, we found a significant difference between task 2 and 3 ($t(31) = 3.76$, $p < .01$) with medium effect size. This was not reflected in the time participants needed to accomplish the tasks: Both took a statistically similar amount of time, $M = 6.48$ min ($SD = 2.40$) for task 2, and $M = 5.53$ min ($SD = 3.48$) for task 3, with a rather small effect size ($d = .32$).

In general, as shown in Table 1, interaction adequacy was assessed equally positive for both tasks (small effect size). When asked specifically whether they understood how their interactions affected the landscape, participants seemed also satisfied, in task 2 ($M = 4.00$, $SD = 0.76$) and in task 3 ($M = 3.44$, $SD = 1.24$), without significant difference ($d = .54$).

Also the results for the constructs mentioned before, e.g. overall satisfaction, system effectiveness, perceived usefulness and recommendation quality, show a positive assessment of the interaction possibilities for manipulating a preference profile.

Usability and user experience: Usability of our demonstrator was evaluated as *good* with a SUS-score of 75. On the different scales of the UEQ, we received promising results (ranging from 0.84 to 2.10), in particular, for perspicuity (1.66, *good*), stimulation (1.56, *excellent*) and novelty (2.10, *excellent*).

Demographics and domain knowledge: Participants generally stated that they love movies ($M = 3.91$, $SD = 0.78$), and 75 % reported that they regularly use sites like *IMDb* or *Rotten Tomatoes* for searching further information. Participants were not very familiar with standard 3D applications ($M = 2.25$, $SD = 0.62$), e.g. *Google Earth* or 3D computer games. The expertise with professional 3D applications such as *3DS Max* was, as expected, even lower ($M = 1.41$, $SD = 0.61$). However, we did not find any noteworthy influence of demographics or domain knowledge on our dependent variables.

Discussion

Overall, the study results suggest that our approach provides users with an easy to understand 3D visualization of item space and preferences. Although we built on model-based CF, which is generally considered to be a rather opaque technique, participants were able to make sense of the generated map and the positioning of items on the map. Relying only on the hidden semantics of latent factors, the initial selection of representative samples appeared to be a good starting point for further exploration. Observed interaction behavior shows that the interaction tools provided, e.g. the possibility to request more items, also contribute to participants quickly getting an overview. Consequently, they were able to successfully accomplish search tasks although the overall number of items in the dataset was large. As expected, our approach performed better in situations with a more general search direction in mind than for known-item search. Looking for concrete items could however easily be supported by providing additional search functionalities. Our study, in contrast, has shown that using a latent factor model without any content information seems especially of value in different, yet very common situations where users are searching with respect to "soft" criteria.

With respect to representation of their preferences, as well as the means provided to establish or manipulate the underlying profile, participants were also satisfied. Using the elevation profile of the map to visualize the user's preferences seemed to be supportive in order to reveal how the user is represented within the system. Furthermore, this way being able to set

	Item	*M*	*SD*
RQ1	The positioning of movies inside the landscape was comprehensible.	3.31	0.90
	The presentation of movies inside the landscape helped me getting an overview of the item space.	3.91	0.93
	The recommender system makes me more aware of my choice options.	3.91	0.93
RQ2	The landscape helped me to understand my user profile within the system.	3.63	1.07
	I think the landscape helped me to understand why the movies have been recommended to me.	3.69	0.93
RQ3	The recommender system allows me to tell what I like/dislike.	3.97	1.12
	The recommender system allows me to modify my taste profile.	3.72	0.96
	I felt to be in control over the recommendation process by manipulating the landscape.	3.69	0.97
General results	The recommender system gave me valuable recommendations.	3.97	0.90
	The recommender system helped me find the ideal item.	4.00	0.76
	I enjoyed using the system very much.†	5.47	1.02
	Using the system was fun to do.†	5.38	1.26

Table 2. Mean values and standard deviations for selected questionnaire items, grouped by our research questions († indicates items assessed using a positive 7-point Likert scale, in all other cases, a positive 5-point Likert scale was used).

recommended items in relation to the rest of the item space, appears to positively influence perceived recommendation transparency. This is reflected by the fact that participants stated to understand why recommendations were shown at certain positions. The moderate significant difference regarding comprehensibility of the landscape between task 2 and 3 is likely a result of initially presenting a profile from another person in the latter case. Since preferences from a completely different profile might still have influenced the results after participants finished the interaction, it seems reasonable that they perceived the landscape as slightly less comprehensible.

Explicit user feedback is often very sparse and motivating users to state their preferences, for example, by means of ratings, is known to be difficult [19, 16]. In this light, it seems particularly promising that participants felt in control over our system while enjoying the interaction. Thus, shaping the landscape by creating hills and valleys appears to be an appropriate means for expressing preferences. This is also supported by the positive results in terms of interaction adequacy, usability, and user experience. The significant difference between task 2 and 3 with respect to perceived effort may again be ascribed to the fact that participants had to manipulate an existing profile in the latter case, which is likely to be a more complex task than starting from a flat surface. In addition, the continuous updates of landscape and recommendations in task 3 may also have contributed to perceiving the effort to be slightly higher. Thus, although the scores are still in a satisfactory range for both tasks, further investigation will be needed to account for the different task settings. Either way, participants were satisfied with the resulting recommendations, both when they started to establish a preference profile as well as when they had to manipulate an existing one.

CONCLUSIONS AND OUTLOOK

To answer the research questions posed at the beginning of this paper, we introduced a novel 3D visualization with a landscape of hills and valleys in order to represent a large item space, show the user's preferences in that space, and allow him or her to manipulate the underlying model. We implemented a demonstrator that indicates the usefulness of our approach—also in cold-start situations. In the user study we conducted, we obtained promising results concerning our research questions, and especially regarding perceived transparency, recommendation quality, user enjoyment, and degree of control users are able to exert over the system.

Apparently, a latent factor model inferred by MF from ratings as they are customary in CF may not only serve to calculate accurate recommendations, but also conveys semantics that can be revealed to the user. While this is in line with earlier research [53, 13], we show that latent factors may be a legitimate source for positioning a large number of items on a map that users perceive as comprehensible. Without requiring any content-related data, preferences can both be presented and successfully elicited with respect to regions of the item space the user is particularly interested in—independent of knowing and rating specific items. Although MF-based methods are typically intransparent due to their statistical nature, our study suggests that using a modern visualization technique together with representative sample items, supports users in understanding the representation of their preferences within the system, i.e. the user model, and the resulting recommendations.

Despite the potential shown by our interactive recommending approach based on conventional model-based CF, it is in principle independent of algorithms and background data. In future work, we therefore aim at using recommender algorithms other than MF, mapping and sampling techniques besides MDS and k-means, and also further datasources, e.g. content information instead of or in addition to ratings. This goes along with our goal of implementing the approach in a different domain or in a cross-domain scenario, where the need to deal with more heterogeneous data as well as a larger number of items is even more apparent. Furthermore, there is room left for improvement with respect to the visualization and interaction concept. For instance, additional samples could immediately be shown when zooming in. Also, the usage of a map metaphor may be further exploited, e.g. by highlighting regions on the map and labeling them with tags. In general, one can think of using entirely different interaction mechanisms or even a tangible user interface. Finally, while the present user study focused on a proof-of-concept, we are also interested in conducting more in-depth comparisons, in particular with a baseline system as well as other state-of-the-art interactive RS and visualizations.

REFERENCES

1. Jae-Wook Ahn and Peter Brusilovsky. 2013. Adaptive Visualization for Exploratory Information Retrieval. *Information Processing & Management* 49, 5 (2013), 1139–1164.
2. Xavier Amatriain and Josep M. Pujol. 2015. *Recommender Systems Handbook*. Springer US, Chapter Data Mining Methods for Recommender Systems, 227–262.
3. Xavier Amatriain, Josep M. Pujol, Nava Tintarev, and Nuria Oliver. 2009. Rate It Again: Increasing Recommendation Accuracy by User Re-rating. In *Proc. RecSys '09*. ACM, 173–180.
4. Fedor Bakalov, Marie-Jean Meurs, Birgitta König-Ries, Bahar Sateli, René Witte, Greg Butler, and Adrian Tsang. 2013. An Approach to Controlling User Models and Personalization Effects in Recommender Systems. In *Proc. IUI '13*. ACM, 49–56.
5. Dmitry Bogdanov, Martín Haro, Ferdinand Fuhrmann, Anna Xambó, Emilia Gómez, and Perfecto Herrera. 2013. Semantic Audio Content-Based Music Recommendation and Visualization Based on User Preference Examples. *Information Processing & Management* 49, 1 (2013), 13–33.
6. Ingwer Borg and Patrick J. F. Groenen. 2005. *Modern Multidimensional Scaling: Theory and Applications* (2 ed.). Springer.
7. Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. 2012. TasteWeights: A Visual Interactive Hybrid Recommender System. In *Proc. RecSys '12*. ACM, 35–42.
8. John Brooke. 1996. SUS – A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry*. Taylor & Francis, 189–194.
9. Li Chen and Pearl Pu. 2012. Critiquing-Based Recommenders: Survey and Emerging Trends. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012), 125–150.
10. Mei C. Chuah. 1998. Dynamic Aggregation with Circular Visual Designs. In *Proc. INFOVIS '98*. IEEE, 35–43.
11. Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. 2012. User Effort vs. Accuracy in Rating-Based Elicitation. In *Proc. RecSys '12*. ACM, 27–34.
12. Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2016a. Tag-Enhanced Collaborative Filtering for Increasing Transparency and Interactive Control. In *Proc. UMAP '16*. ACM, 169–173.
13. Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2016b. Towards Understanding Latent Factors and User Profiles by Enhancing Matrix Factorization with Tags. In *Poster Proc. RecSys '16*.
14. Mehdi Elahi, Francesco Ricci, and Neil Rubens. 2014. Active Learning Strategies for Rating Elicitation in Collaborative Filtering: A System-Wide Perspective. *ACM Transactions on Intelligent Systems and Technology* 5, 1 (2014), 13:1–13:33.
15. Siamak Faridani, Ephrat Bitton, Kimiko Ryokai, and Ken Goldberg. 2010. Opinion Space: A Scalable Tool for Browsing Online Comments. In *Proc. CHI '10*. ACM, 1175–1184.
16. Sebastian Feil, Martin Kretzer, Karl Werder, and Alexander Maedche. 2016. Using Gamification to Tackle the Cold-Start Problem in Recommender Systems. In *CSCW '16 Companion*. ACM, 253–256.
17. Emden Gansner, Yifan Hu, Stephen Kobourov, and Chris Volinsky. 2009. Putting Recommendations on the Map – Visualizing Clusters and Relations. In *Proc. RecSys '09*. ACM, 345–348.
18. Brynjar Gretarsson, John O'Donovan, Svetlin Bostandjiev, Christopher Hall, and Tobias Höllerer. 2010. SmallWorlds: Visualizing Social Recommendations. *Computer Graphics Forum* 29, 3 (2010), 833–842.
19. F. Maxwell Harper, Xin Li, Yan Chen, and Joseph A. Konstan. 2005. An Economic Model of User Rating in an Online Recommender System. In *Proc. UM '05*. Springer, 307–316.
20. John A. Hartigan and M. Anthony Wong. 1979. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979), 100–108.
21. Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive Recommender Systems: A Survey of the State of the Art and Future Research Challenges and Opportunities. *Expert Systems with Applications* 56, 1 (2016), 9–27.
22. Marti Hearst. 2009. *Search User Interfaces*. Cambridge University Press.
23. Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky. 2010. A Tour Through the Visualization Zoo. *ACM Queue* 53, 6 (2010), 59–67.
24. Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining Collaborative Filtering Recommendations. In *Proc. CSCW '00*. ACM, 241–250.
25. Anthony Jameson, Martijn C. Willemsen, Alexander Felfernig, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Li Chen. 2015. *Recommender Systems Handbook*. Springer US, Chapter Human Decision Making and Recommender Systems, 611–648.
26. 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* 4, 2 (2014), 8:1–8:26.
27. Yucheng Jin, Karsten Seipp, Erik Duval, and Katrien Verbert. 2016. Go With the Flow: Effects of Transparency and User Control on Targeted Advertising Using Flow Charts. In *Proc. AVI '16*. ACM, 68–75.

28. Martijn Kagie, Michiel van Wezel, and Patrick J. F. Groenen. 2011. *Recommender Systems Handbook*. Springer, Chapter Map Based Visualization of Product Catalogs, 547–576.

29. Daniel A. Keim and Hans-Peter Kriegel. 1996. Visualization Techniques for Mining Large Databases: A Comparison. *IEEE Transactions on Knowledge and Data Engineering* 8, 6 (1996), 923–938.

30. Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North (Eds.). 2008. *Information Visualization: Human-Centered Issues and Perspectives*. Springer.

31. Mohammad Khoshneshin and W. Nick Street. 2010. Collaborative Filtering via Euclidean Embedding. In *Proc. RecSys '10*. ACM, 87–94.

32. Bart P. Knijnenburg and Martijn C. Willemsen. 2015. *Recommender Systems Handbook*. Springer US, Chapter Evaluating Recommender Systems with User Experiments, 309–352.

33. Bart P. Knijnenburg, Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the User Experience of Recommender Systems. *User Modeling and User-Adapted Interaction* 22, 4-5 (2012), 441–504.

34. Jürgen Koenemann and Nicholas J. Belkin. 1996. A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness. In *Proc. CHI '96*. ACM, 205–212.

35. 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.

36. Yehuda Koren and Robert Bell. 2015. *Recommender Systems Handbook*. Springer US, Chapter Advances in Collaborative Filtering, 77–118.

37. Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

38. Johannes Kunkel, Benedikt Loepp, and Jürgen Ziegler. 2015. 3D-Visualisierung zur Eingabe von Präferenzen in Empfehlungssystemen [3D visualization to elicit preferences in recommender systems]. In *Proc. M&C '15*. De Gruyter Oldenbourg, 123–132.

39. Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *HCI and Usability for Education and Work*. Springer, 63–76.

40. Benedikt Loepp, Catalin-Mihai Barbu, and Jürgen Ziegler. 2016. Interactive Recommending: Framework, State of Research and Future Challenges. In *Proc. EnCHIReS '16*.

41. Benedikt Loepp, Katja Herrmanny, and Jürgen Ziegler. 2015. Blended Recommending: Integrating Interactive Information Filtering and Algorithmic Recommender Techniques. In *Proc. CHI '15*. ACM, 975–984.

42. Benedikt Loepp, Tim Hussein, and Jürgen Ziegler. 2014. Choice-Based Preference Elicitation for Collaborative Filtering Recommender Systems. In *Proc. CHI '14*. ACM, 3085–3094.

43. Sean M. McNee, Shyong K. Lam, Joseph A. Konstan, and John Riedl. 2003. Interfaces for Eliciting New User Preferences in Recommender Systems. In *Proc. UM '03*. Springer, 178–187.

44. Afshin Moin. 2014. A Unified Approach to Collaborative Data Visualization. In *Proc. SAC '14*. ACM, 280–286.

45. Chris Muelder, Thomas Provan, and Kwan-Liu Ma. 2010. Content Based Graph Visualization of Audio Data for Music Library Navigation. In *Proc. ISM '10*. IEEE, 129–136.

46. Sayooran Nagulendra and Julita Vassileva. 2014. Understanding and Controlling the Filter Bubble Through Interactive Visualization: A User Study. In *Proc. HT '14*. ACM, 107–115.

47. Eli Pariser. 2011. *The Filter Bubble: What the Internet is Hiding From You*. Penguin Press.

48. Souneil Park, Seungwoo Kang, Sangyoung Chung, and Junehwa Song. 2009. NewsCube: Delivering Multiple Aspects of News to Mitigate Media Bias. In *Proc. CHI '09*. ACM, 443–452.

49. Denis Parra, Peter Brusilovsky, and Christoph Trattner. 2014. See What You Want to See: Visual User-Driven Approach for Hybrid Recommendation. In *Proc. IUI '14*. ACM, 235–240.

50. Pearl Pu, Li Chen, and Rong Hu. 2011. A User-Centric Evaluation Framework for Recommender Systems. In *Proc. RecSys '11*. ACM, 157–164.

51. Pearl Pu, Li Chen, and Rong Hu. 2012. Evaluating Recommender Systems from the User's Perspective: Survey of the State of the Art. *User Modeling and User-Adapted Interaction* 22, 4-5 (2012), 317–355.

52. Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). 2015. *Recommender Systems Handbook* (2 ed.). Springer US.

53. Marco Rossetti, Fabio Stella, and Markus Zanker. 2013. Towards Explaining Latent Factors with Topic Models in Collaborative Recommender Systems. In *Proc. DEXA '13*. 162–167.

54. Richard M. Ryan. 1982. Control and Information in the Intrapersonal Sphere: An Extension of Cognitive Evaluation Theory. *Journal of Personality and Social Psychology* 43, 3 (1982), 450–461.

55. Amit Sharma and Dan Cosley. 2013. Do Social Explanations Work? Studying and Modeling the Effects of Social Explanations in Recommender Systems. In *Proc. WWW '13*. ACM, 1133–1144.

56. Ben Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc. VL '96*. IEEE, 336–343.

57. Rashmi Sinha and Kirsten Swearingen. 2002. The Role of Transparency in Recommender Systems. In *CHI '02 Extended Abstracts*. ACM, 830–831.

58. E. Isaac Sparling and Shilad Sen. 2011. Rating: How Difficult is it?. In *Proc. RecSys '11*. ACM, 149–156.

59. Pieter Jan Stappers, Gert Pasman, and Patrick J. F. Groenen. 2000. Exploring Databases for Taste or Inspiration with Interactive Multi-Dimensional Scaling. *Proc. HFES '00* (2000), 575–578.

60. Nava Tintarev and Judith Masthoff. 2015. *Recommender Systems Handbook*. Springer US, Chapter Explaining Recommendations: Design and Evaluation, 353–382.

61. James Uther and Judy Kay. 2003. VlUM, a Web-Based Visualisation of Large User Models. In *Proc. UM '03*. Springer, 198–202.

62. Katrien Verbert, Denis Parra, Peter Brusilovsky, and Erik Duval. 2013. Visualizing Recommendations to Support Exploration, Transparency and Controllability. In *Proc. IUI '13*. ACM, 351–362.

63. Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: Explaining Recommendations Using Tags. In *Proc. IUI '09*. ACM, 47–56.

64. Jesse Vig, Shilad Sen, and John Riedl. 2011. Navigating the Tag Genome. In *Proc. IUI '11*. ACM, 93–102.

65. David Wong, Siamak Faridani, Ephrat Bitton, Björn Hartmann, and Ken Goldberg. 2011. The Diversity Donut: Enabling Participant Control over the Diversity of Recommended Responses. In *CHI '11 Extended Abstracts*. ACM, 1471–1476.

66. Bo Xiao and Izak Benbasat. 2007. E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact. *MIS Quarterly* 31, 1 (2007), 137–209.