Explorative Analysis of Recommendations Through Interactive Visualization

Christian Richthammer and Günther Pernul

Department of Information Systems
University of Regensburg, Regensburg, Germany
{firstname.lastname}@wiwi.uni-regensburg.de

Abstract. Even though today's recommender algorithms are highly sophisticated, they can hardly take into account the users' situational needs. An obvious way to address this is to initially inquire the users' momentary preferences, but the users' inability to accurately state them upfront may lead to the loss of several good alternatives. Hence, this paper suggests to generate the recommendations without such additional input data from the users and let them interactively explore the recommended items on their own. To support this explorative analysis, a novel visualization tool based on treemaps is developed. The analysis of the prototype demonstrates that the interactive treemap visualization facilitates the users' comprehension of the big picture of available alternatives and the reasoning behind the recommendations. This helps the users get clear about their situational needs, inspect the most relevant recommendations in detail, and finally arrive at informed decisions.

Key words: Recommender systems, interactive visualization, search space, explorative analysis

1 Introduction

Nowadays, the ever-increasing amounts of alternatives and information overwhelm people in their decision making processes. Recommender systems, which emerged in the mid 1990s [1, 2, 3] and formed a research field of their own since then, constitute an important approach to address this phenomenon of information overload [4, 5]. Their main aim is to inform users about items they will probably like and have previously not been aware of. This is also referred to as the recommendation problem [1, 4]. For many years, the dominating topic in the recommender systems community has been the development of prediction algorithms along with the evaluation of their recommendation accuracy [5, 6], experiencing its peak with the Netflix Prize [7]. As the development of new algorithms and the optimization of existing ones often result in only marginal improvements by now [8, 9], a shift of research interests to the ultimate user experience can be observed [10]. Researchers have realized that algorithms may be the backbone of any good recommender system but that the users' evaluation of the system and its recommendations is crucial [11, 12].

Traditional recommender algorithms can, even in their sophisticatedly optimized form, only produce a set of recommendations that may generally be of interest to the users. Considering that the users may have very special needs (notably deviating from their preferences determined according to their past rating behaviors) in a particular situation, the top recommendation does not necessarily have to be the best choice in that situation. An obvious way to address this is to initially inquire the users' situational needs as done in knowledge-based recommender systems [13]. The users' input can then be used for filtering or weighting processes. However, the users are unlikely to be able to accurately state all of their preferences upfront, especially without having seen any suggestions and if there are any suitable alternatives fulfilling their criteria at all [9, 14]. Thus, filtering may lead to the loss of several alternatives that are worth considering but lie just outside the defined filtering criteria.

In contrast to the aforementioned initial inquiry of the users' situational needs, this paper suggests to generate the set of recommendations without such additional input data from the users and let them interactively explore the recommended items on their own with their situational needs in mind. We aim to support this explorative analysis by relying on interactive visualization, a concept that has been ascribed a lot of uncovered potential in the field of recommender systems [8, 15]. In particular, we develop a novel visualization tool based on treemaps. Treemaps arrange the recommendations in a way that provides a structured overview of large parts of the search space (i.e. the set of all available alternatives), which facilitates the users' comprehension of the big picture of options fitting their needs [16]. On the one hand, educating the users about the search space helps them understand the reasoning behind the recommendations [17]. On the other hand, it serves the users as a basis to get clear about their situational needs, reduce the number of displayed recommendations to the top ones, and inspect these in detail.

The remainder of the paper is structured according to the guidelines for conducting design science research by Hevner et al. [18]. In particular, it follows the six activities of the design science research methodology introduced by Peffers et al. [19]. In this section, the first activity (*problem identification and motivation*) is covered by introducing the problem context and expounding the motivation for the explorative analysis of recommendations. In Section 2, related work on visualizations for recommender systems are discussed and the objectives of our solution are pointed out as demanded by the second activity (*define the objectives for a solution*). The third activity (*design and development*) is split into two steps: presenting the conceptual design of the treemap visualization in Section 3, and describing its prototypical implementation in Section 4. Section 5 is dedicated to the demonstration of our solution and the evaluation of our research objectives, thus covering the fourth (*demonstration*) and fifth activity (*evaluation*). The discussion of aspects for future work in Section 6 concludes the paper. Publishing our findings covers the sixth activity (*communication*).

2 Related Work and Research Objectives

The research interest in visualizations of recommendations that are more sophisticated than simple lists has been growing steadily since the end of the last decade. One of the first notable proposals in this area is PeerChooser [20], whose graph-based interface is supposed to provide a visual explanation of the collaborative filtering process and to enable the users to interactively manipulate their neighborhood. Also using a graph-based interface, SmallWorlds [21] allows to express item preference profiles with which personalized item recommendations are generated in a transparent manner. This enables the users both to explore their relationships to their peers and to discover new relevant items. Exploring relationships to peers is also the motivation behind SFViz [22]. Here, friend recommendations in social networks are visualized using a radial space-filling technique as well as a circle layout with edge bundling. Residing in a similar application area, TasteWeights [23] depicts item predictions based on social web resources (e.g. Wikipedia, Facebook, and Twitter). Its main characteristic is that the users are able to interactively adjust the weights of the different context sources and the corresponding items. Assigning weights to different recommendation strategies is an important feature in SetFusion [24], too.

In recent years, a number of researchers have used map-based interfaces to support the users in understanding and exploring recommendations. In particular, Gansner et al. [25] use geographic maps for TV show recommendations, Kagie et al. [26] use two-dimensional maps for product recommendations in e-commerce, and Verbert et al. [27] use clustermaps for talk recommendations at scientific conferences. Specifically concentrating on treemaps, Tintarev and Masthoff [16, 28] state that this kind of visualization has not been used for recommender systems even though it may be a valuable choice. Based on our literature search, we can verify this claim and confirm that it still holds. The only proposal going into this direction is the treemap-like component presented by Katarya et al. [29], which is included in their interface containing several visual explanation styles for movie recommendations. We call it “treemap-like” because we are missing two distinctive properties of treemaps (cf. Section 3). First, the visualization does not use the available space to its full extent. And second, it does not display any hierarchical structures. In this work, we employ the treemap visualization in its traditional form because we believe these two properties to be important for depicting the search space as completely as possible. Furthermore, the focus of Katarya et al. [29] differs from ours. They concentrate on different ways to visualize explanations for recommendations, whereas explanations are rather an add-on in our proposal. We believe that a good visualization can improve user satisfaction by itself, which is why we lay our primary focus on the detailed examination of the treemap.

The analysis of related work shows that despite the generally increasing interest in interactive visualizations for recommender systems, the employment of treemaps has received comparatively little attention so far. Since this kind of visualization has proven to be effective in connection with non-personalized news recommendations [30], we want to explore this promising concept in its traditional form for personalized recommendations as well. The overall goal of our treemap visualization is to support the users' explorative analysis of recommendations. In particular, we want to achieve this by pursuing the following objectives:

1. Facilitate the users' comprehension of the recommendations by educating them about the search space.
2. Enable the users to interactively explore the recommendations with their situational needs in mind.
3. Increase the transparency of the recommender system by helping the users understand the accomplishment of the recommendations.

3 Conceptual Design

The treemap (first designed in the early 1990s [31, 32]) is a space-constrained visualization that uses nested rectangles to display large amounts of monohierarchical, tree-structured data with great efficiency and high readability [33]. The sizes of the rectangles are calculated by the underlying tiling algorithm, for which several popular alternatives with different characteristics are available. Moreover, it is possible to use different colors and different degrees of opacity for the rectangles, which enables displaying additional attributes. Ways to display even more attributes include different font sizes, font styles, and borders of the rectangles. However, we limit the employment of dimensions to the following four because we do not want to clutter our treemap visualization.

Hierarchy. As its name suggests and as already pointed out, the treemap is used to visualize tree-structured data. In contrast to the size dimension, the overarching hierarchy dimension is not used for quantitative data but for distinct categories (e.g. different states when visualizing voting data of US presidential elections).

Size. The size dimension determines the size of the single rectangles. It should represent a quantifiable attribute so that the sizes of the rectangles at a lower hierarchical level can be summed up to the size of the superordinate rectangle. The largest rectangle is arranged in the top left corner.

Color. The color dimension determines the color of the single rectangles. It is most qualified to depict a measure of performance, change, or development [34]. However, it can also be used to support the visualization of the hierarchy dimension by assigning different colors to the different elements on a certain hierarchy level.

Opacity. The opacity dimension can be used in addition to the color dimension and in order to reduce the degree of visibility of items that are less relevant according to a specified criterion. In a news recommender system, for example, older stories may be less interesting than more recent stories regardless of their overall predicted importance.

Figure 1 provides a conceptual overview of our visualization component including the dimensions introduced before. As can be seen in the bottom part of the figure, we want the users to be able to change the assignment of the available attributes to the dimensions in order to give them full control over the visualization. In addition, the users are provided with comprehensive information on the items in the form of a tooltip. This includes general metadata (e.g. title, artist, and release year in a music recommender) but also recommendation-specific data such as predicted rating, overall rating, and further explanations. When clicking on a treemap element, the corresponding tooltip is expanded (cf. Figure 4).

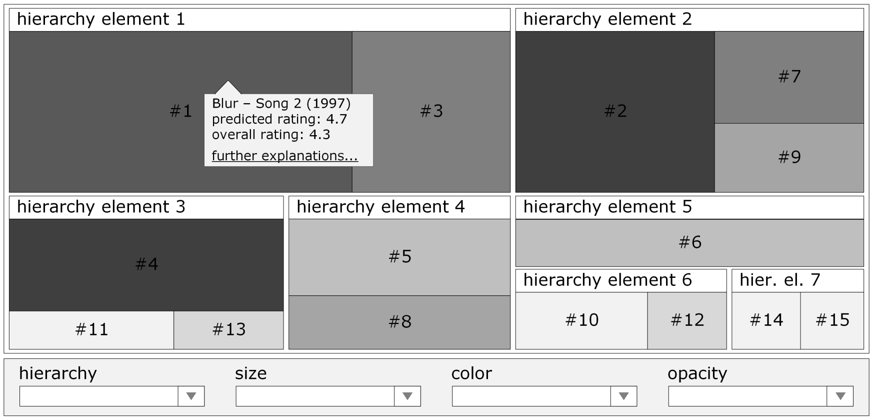


Fig. 1. Conceptual overview of the treemap visualization component.

Although this proposal's focus is on the pure visualization, we include explanations as an add-on to further increase the user acceptance of recommendations [35]. The positive effect on the user acceptance mainly stems from the increase in transparency that follows from the employment of explanations. When providing explanations, recommender systems are no black boxes anymore. Instead, they feel like the digital analogue to the transparent social process of recommendation-making in the real world [36]. Specifically, we add a histogram (cf. Figure 4) showing the ratings of other users to each item because this easy-to-understand form of explanation performed especially well in Herlocker et al.'s [37] pioneering study on the persuasiveness of explanations. In contrast to this, we also provide detailed information on the calculations and intermediate values of the algorithms such as the similarity values of the nearest neighbors. Because of the complexity of the latter form of explanation, the corresponding information is collapsed by default but can be expanded by interested users (cf. Figure 4).

4 Implementation

In this section, we demonstrate the feasibility of the conceptual design by implementing it in a software prototype. We base the implementation on the popular

MovieLens 100K dataset (100,000 ratings from 943 users on 1,682 movies) [38] because its list of entities and attributes perfectly fits our conceptual design:

– rating: user ID, movie ID, rating, timestamp
– user: user ID, age, gender, occupation, zip code
– movie: movie ID, title, release date, IMDb URL, genre

The release date, the genre, the popularity (number of ratings), and the average rating of the movies can be employed as attributes for the treemap dimensions introduced before. The age, the gender, the occupation, and the zip code of the users can be employed for the explanation of user-based collaborative filtering algorithms. The IMDb URL is needed to provide additional data on the movies. The list of attributes is completed by the predicted rating of the movies, which is determined by the recommender algorithm and used as another attribute for the treemap dimensions.

Figure 2 provides an overview of the architecture underlying our prototype. In the backend, we use the LAMP solution stack, which consists of a Linux operating system, an Apache web server, a MySQL database management system, and the PHP programming language. The frontend of the prototype is realized using HTML5, CSS, and the JavaScript framework jQuery. Thus, it is platform-independent and can easily be accessed through a web browser.

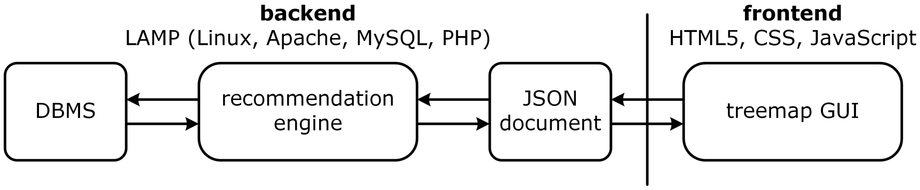


Fig. 2. Application architecture of the treemap prototype.

When the users trigger the generation of a new set of recommendations or the alteration of the currently displayed treemap, the parameters of their request are transferred to the recommendation engine in the form of a JSON document. The parameters mainly include the assignment of the attributes to the treemap dimensions (cf. Section 3) as well as the recommender algorithm along with the corresponding settings. Regarding the recommender algorithms, the prototype relies on the Vogoo PHP library[1], whose user-based collaborative filtering, item-based collaborative filtering and SlopeOne implementations are used in slightly adjusted forms. The recommendation engine collects the data needed to process the request from the database management system (DBMS), generates the set of recommendations, and forwards them to the treemap graphical user intergace (GUI) in the form of a JSON document. Finally, this JSON document is employed to create the treemap, which is realized using the JavaScript libraries D3.js and D3plus.

[1] https://sourceforge.net/projects/vogoo/

5 Descriptive Evaluation

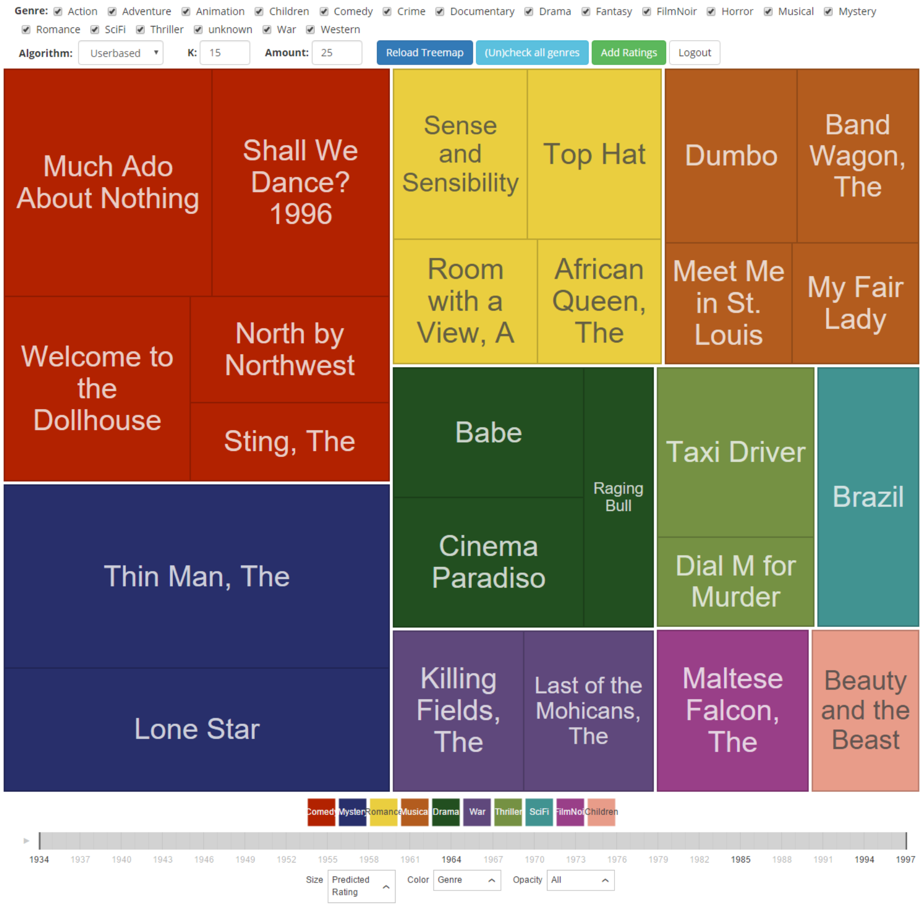
To demonstrate the proper functioning and goodness of our solution, we carry out a descriptive scenario-based evaluation. In particular, we present a realistic scenario with which we show how our prototype can enhance the exploration of recommendations in practice. To construct the scenario, the prototype has been extended with the functionality to add new users and ratings to the dataset. Due to space restrictions, however, these procedures are not discussed in detail. Instead, the reader is referred to the following URL created for demonstration purposes: `http://rec-ifs.uni-r.de/treemap/` (Note that the debug mode allows to use the tool from the perspective of each of the MovieLens 100K users.)

5.1 Scenario Analysis

Suppose it is Friday evening and Alice wants to watch a movie together with her boyfriend Bob. In general, Alice loves romances and likes comedies and dramas. She dislikes western, sci-fi and crime movies. Considering the presence of Bob, Alice is quite sure that she wants to receive a recommendation for a comedy or a drama for tonight. A romance would be conceivable only if it was a highly popular and generally liked one. In addition, any movie with a very high predicted rating for Alice might be a good choice as long as it is not from one of her disliked genres. Applying a genre filter before having seen any recommendations may exclude these other good alternatives simply because Alice is not aware of them. Hence, Alice decides to apply no genre filter at all before not having explored the search space with the help of our treemap visualization.

After Alice has provided the recommender system with several movie ratings, the treemap is generated using the default algorithm settings (user-based collaborative filtering, 15 nearest neighbors, 25 displayed recommendations) as depicted in Figure 3. Because of the clustering of the movies by genre and the clearly depicted ratios between them, Alice quickly gets an impression of the relevance of the different genres. As expected from her personal preferences, romances are comparatively important. However, comedies and mystery movies are even more important in the search space than romances. Looking at single movies, the two mystery movies "Thin Man, The" and "Lone Star" as well as the comedies "Much Ado About Nothing" and "Shall We Dance? 1996" are particularly conspicuous because of the large size of the corresponding rectangles. As a result of this first overview of the search space, Alice now knows that especially comedies, romances, and mystery movies come into question for tonight.

With this insight in mind, it makes sense for Alice to compare the recommendations in more detail by applying different filters and displaying different attributes through the treemap dimensions. Hereto, she selects only "Comedy", "Mystery", and "Romance" in the genre filter above the treemap visualization. She also reduces the number of recommendations to 10, which makes the size differences of the rectangles even more obvious. Using the opacity dimension to visualize the release years of the movies reveals that "Thin Man, The" is from



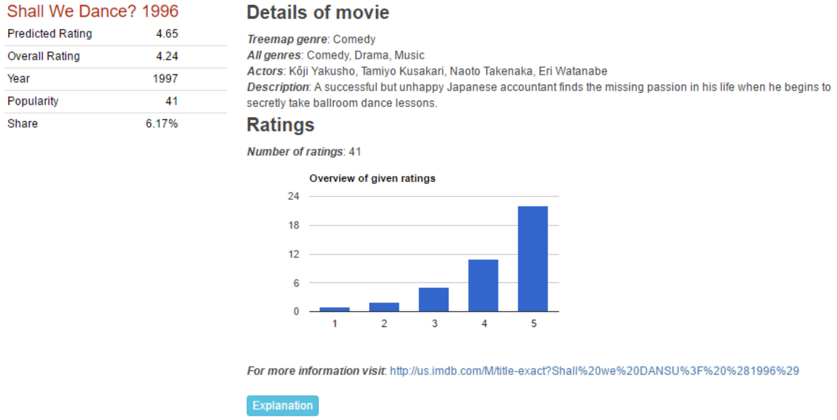
Genre: ☑ Action ☑ Adventure ☑ Animation ☑ Children ☑ Comedy ☑ Crime ☑ Documentary ☑ Drama ☑ Fantasy ☑ FilmNoir ☑ Horror ☑ Musical ☑ Mystery ☑ Romance ☑ SciFi ☑ Thriller ☑ unknown ☑ War ☑ Western

Algorithm: Userbased **K:** 15 **Amount:** 25

Size: Predicted Rating Color: Genre Opacity: All

Fig. 3. Screenshot of the software prototype.

1934. Although this movie has the highest predicted rating among the recommendations, Alice realizes that she prefers a more recent movie for tonight and decides to exclude older ones from her decision making. Hence, she sets the lower bound of the timeline filter below the treemap visualization to 1988. By visualizing the movies' overall ratings through the color dimension, Alice discovers that the comedy "Shall We Dance? 1996" has both a high overall and high predicted rating and would therefore be a suitable candidate. However, the tooltip information corresponding to the movie shows that its theme is very musical (cf. Figure 4), which is something Bob does not like at all. The remaining alternatives with a very high predicted rating for Alice are the comedy "Much Ado About Nothing" and the mystery movie "Lone Star". As they do not show any notable differences regarding their attributes, Alice finally chooses "Much Ado About Nothing" because the tooltip information mentions that it is based on the identically named play by Shakespeare, which perfectly fits her preferences.



Shall We Dance? 1996

Predicted Rating	4.65
Overall Rating	4.24
Year	1997
Popularity	41
Share	6.17%

Details of movie

Treemap genre: Comedy
All genres: Comedy, Drama, Music
Actors: Kôji Yakusho, Tamiyo Kusakari, Naoto Takenaka, Eri Watanabe
Description: A successful but unhappy Japanese accountant finds the missing passion in his life when he begins to secretly take ballroom dance lessons.

Ratings

Number of ratings: 41

For more information visit: http://us.imdb.com/M/title-exact?Shall%20we%20DANSU%3F%20%281996%29

Fig. 4. Screenshot of the tooltip corresponding to "Shall We Dance? 1996".

5.2 Evaluation of Research Objectives

In summary, the scenario analysis demonstrates several major benefits of our visualization prototype, which are directly related to the accomplishment of the objectives pointed out in Section 2. First and most important, the users' comprehension of the recommendations is facilitated by educating them about the search space (cf. first objective). On the one hand, the structured initial overview of the best alternatives in the search space clustered by categories prevents the users from missing out on very good alternatives because of a rash usage of a category filter. In this example, Alice would have never been aware of the two highly recommended mystery movies "Thin Man, The" and "Lone Star" if she had limited her recommendations to romances, comedies, and dramas from the beginning. On the other hand, the overview of the search space – especially through the size dimension – helps the users get a quick impression of the amount of alternatives with a very high predicted rating. This insight can be used to reduce the number of simultaneously displayed recommendations, so that a more detailed analysis is limited to the ones that actually come into question.

Second, the users can interactively explore the recommendations with their situational needs in mind (cf. second objective). In addition to providing a structured overview, the treemap (along with its dimensions) enables the comparison of multiple attributes at once without reducing the visualization's readability. In the scenario, for example, we mention the simultaneous depiction of three attributes: the genre through hierarchy, the predicted rating through size, and the overall rating through color.

Finally, although not specific to our treemap visualization, the transparency of the recommender system is increased (cf. third objective) by providing the users with as many information available to the recommender system as possible, which includes both raw data from the dataset and intermediate values from the recommendation process. Thus, the users are supported in understanding the accomplishment of their recommendations.

6 Conclusion and Future Work

Even though today's recommender algorithms are highly sophisticated, they can hardly take into account the users' momentary situational needs. This is especially the case if these needs deviate notably from the users' normal preferences determined according to their past rating behaviors. An obvious way to address this is to initially inquire the users' situational needs. However, this may lead to the loss of several good alternatives because the users are unlikely to be able to accurately state all of their preferences upfront. Against this background, this paper presented a treemap visualization that does not require such additional input from the users and instead supports them in the explorative analysis of the recommended items with their situational needs in mind. In the descriptive scenario-based evaluation, we showed that the structured overview provided by the treemap could facilitate the users' comprehension of the big picture of alternatives available in the search space. This helps them understand the reasoning behind the recommendations and serves them as a basis to get clear about their situational needs, inspect the most relevant recommendations in detail, and finally arrive at informed decisions.

In future work, further insights might be gained by using the treemap visualization in application areas other than movies (e.g. books, music) and also with content-based algorithms. Moreover, it would be possible to extend our visualization tool with findings from related recommender systems research areas (e.g. active learning [39], critiquing [40], and more extensive explanations).

Acknowledgments. The research leading to these results was supported by the "Bavarian State Ministry of Education, Science and the Arts" as part of the FORSEC research association. The authors would like to thank Kilian Müller and Regina Staller for the implementation of the prototype used in this paper.

References

1. Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and Evaluating Choices in a Virtual Community of Use. In: Proc. of the 13th SIGCHI Conference on Human Factors in Computing Systems (CHI). (1995) 194–201
2. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW). (1994) 175–186
3. Resnick, P., Varian, H.R.: Recommender Systems. Communications of the ACM **40**(3) (1997) 56–58
4. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering **17**(6) (2005) 734–749
5. Konstan, J.A., Riedl, J.: Recommender Systems: From Algorithms to User Experience. User Modeling and User-Adapted Interaction **22**(1-2) (2012) 101–123
6. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems **22**(1) (2004) 5–53

7. Netflix: The Netflix Prize Rules (2009) [Online]. Available: http://www.netflixprize.com/rules, accessed: April 19, 2016.
8. Loepp, B., Herrmanny, K., Ziegler, J.: Blended Recommending: Integrating Interactive Information Filtering and Algorithmic Recommender Techniques. In: Proc. of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI). (2015) 975–984
9. Pu, P., Chen, L., Hu, R.: 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
10. Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H., Newell, C.: Explaining the User Experience of Recommender Systems. User Modeling and User-Adapted Interaction **22**(4-5) (2012) 441–504
11. McNee, S.M., Riedl, J., Konstan, J.A.: Being Accurate Is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In: CHI '06 Extended Abstracts on Human Factors in Computing Systems. (2006) 1097–1101
12. Swearingen, K., Sinha, R.: Beyond Algorithms: An HCI Perspective on Recommender Systems. In: Proc. of the ACM SIGIR Workshop on Recommender Systems. (2001)
13. Burke, R.: Knowledge-based Recommender Systems. In Kent, A., ed.: Encyclopedia of Library and Information Science: Volume 69 - Supplement 32. Marcel Dekker, Inc., New York, NY, USA and Basel, Schweiz (2000) 180–200
14. Burke, R., Hammond, K., Young, B.: Knowledge-based Navigation of Complex Information Spaces. In: Proc. of the 13th National Conference on Artificial Intelligence (AAAI). (1996) 462–468
15. Herrmanny, K., Schering, S., Berger, R., Loepp, B., Günter, T., Hussein, T., Ziegler, J.: MyMovieMixer: Ein hybrider Recommender mit visuellem Bedienkonzept. In: Mensch und Computer 2014 - Tagungsband. (2014) 45–54
16. Tintarev, N., Masthoff, J.: A Survey of Explanations in Recommender Systems. In: Proc. of the 23rd International Conference on Data Engineering Workshop (ICDE). (2007) 801–810
17. Smyth, B.: Case-Based Recommendation. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web. Volume 4321 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg (2007) 342–376
18. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly **28**(1) (2004) 75–105
19. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems **24**(3) (2007) 45–77
20. O'Donovan, J., Smyth, B., Gretarsson, B., Bostandjiev, S., Höllerer, T.: PeerChooser: Visual Interactive Recommendation. In: Proc. of the 26th SIGCHI Conference on Human Factors in Computing Systems (CHI). (2008) 1085–1088
21. Gretarsson, B., O'Donovan, J., Bostandjiev, S., Hall, C., Höllerer, T.: SmallWorlds: Visualizing Social Recommendations. Computer Graphics Forum **29**(3) (2010) 833–842
22. Gou, L., You, F., Guo, J., Wu, L., Zhang, X.: SFViz: Interest-based Friends Exploration and Recommendation in Social Networks. In: Proc. of the 2011 International Symposium on Visual Information Communication (VINCI). (2011) 1–10
23. Bostandjiev, S., O'Donovan, J., Höllerer, T.: TasteWeights: A Visual Interactive Hybrid Recommender System. In: Proc. of the 6th ACM Conference on Recommender Systems (RecSys). (2012) 35–42

24. Parra, D., Brusilovsky, P., Trattner, C.: See What You Want to See: Visual User-driven Approach for Hybrid Recommendation. In: Proc. of the 19th International Conference on Intelligent User Interfaces (IUI). (2014) 235–240
25. Gansner, E., Hu, Y., Kobourov, S., Volinsky, C.: Putting Recommendations on the Map: Visualizing Clusters and Relations. In: Proc. of the 3rd ACM Conference on Recommender Systems (RecSys). (2009) 345–348
26. Kagie, M., van Wezel, M., Groenen, P.J.: Map Based Visualization of Product Catalogs. In Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., eds.: Recommender Systems Handbook. Springer US, Boston, MA, USA (2011) 547–576
27. Verbert, K., Parra, D., Brusilovsky, P., Duval, E.: Visualizing Recommendations to Support Exploration, Transparency and Controllability. In: Proc. of the 18th International Conference on Intelligent User Interfaces (IUI). (2013) 351–362
28. Tintarev, N., Masthoff, J.: Designing and Evaluating Explanations for Recommender Systems. In Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., eds.: Recommender Systems Handbook. Springer US, Boston, MA, USA (2011) 479–510
29. Katarya, R., Jain, I., Hasija, H.: An Interactive Interface for Instilling Trust and Providing Diverse Recommendations. In: Proc. of the International Conference on Computer and Communication Technology (ICCCT). (2014) 17–22
30. Weskamp, M.: Newsmap (2010) [Online]. Available: http://newsmap.jp/, accessed: April 19, 2016.
31. Johnson, B., Shneiderman, B.: Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. In: Proc. of the 2nd Conference on Visualization (VIS). (1991) 284–291
32. Shneiderman, B.: Tree Visualization with Tree-Maps: 2-D Space-Filling Approach. ACM Transactions on Graphics **11**(1) (1992) 92–99
33. Gasteier, K., Krois, K., Hrdina, F.: Exploratory Search and Content Discovery: The Semantic Media Browser (SMB). In: Proc. of the 13th International Conference on Knowledge Management and Knowledge Technologies (i-Know). (2013) 1–8
34. Gemignani, Z.: 10 Lessons in Treemap Design (2009) [Online]. Available: http://www.juiceanalytics.com/writing/10-lessons-treemap-design, accessed: April 19, 2016.
35. Vig, J., Sen, S., Riedl, J.: Tagsplanations: Explaining Recommendations Using Tags. In: Proc. of the 14th International Conference on Intelligent User Interfaces (IUI). (2009) 47–56
36. Sinha, R., Swearingen, K.: The Role of Transparency in Recommender Systems. In: Extended Abstracts on Human Factors in Computing Systems (CHI EA). (2002) 830–831
37. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining Collaborative Filtering Recommendations. In: Proc. of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW). (2000) 241–250
38. Harper, F.M., Konstan, J.A.: The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems **5**(4) (2016) 1–19
39. Rubens, N., Elahi, M., Sugiyama, M., Kaplan, D.: Active Learning in Recommender Systems. In Ricci, F., Rokach, L., Shapira, B., eds.: Recommender Systems Handbook. Springer US, Boston, MA (2015) 809–846
40. Chen, L., Pu, P.: Critiquing-based Recommenders: Survey and Emerging Trends. User Modeling and User-Adapted Interaction **22**(1-2) (2012) 125–150