A Conversational Collaborative Filtering Approach to Recommendation

Eoin Hurrell[1] and Alan F. Smeaton[1,2]

[1]CLARITY: Centre for Sensor Web Technologies and School of Computing,
[2]INSIGHT: Big Data and Analytics Research Centre and School of Computing,
Dublin City University, Glasnevin, Dublin 9, Ireland.
alan.smeaton@dcu.ie

Abstract. Recent work has shown the value of treating recommendation as a conversation between user and system, which conversational recommenders have done by allowing feedback like "not as expensive as this" on recommendations. This allows a more natural alternative to content-based information access. Our research focuses on creating a viable conversational methodology for collaborative-filtering recommendation which can apply to any kind of information, especially visual. Since collaborative filtering does not have an intrinsic understanding of the items it suggests, i.e. it doesn't understand the content, it has no obvious mechanism for conversation. Here we develop a means by which a recommender driven purely by collaborative filtering can sustain a conversation with a user and in our evaluation we show that it enables finding multimedia items that the user wants without requiring domain knowledge.

1 Introduction

Information retrieval is a pursuit that people have been following since before the use of the Internet but with the arrival of the rich information sources that the Internet brings information retrieval is now a far more commonplace activity. Approaches to information retrieval are multi-faceted and complex and there are multiple tools available which can support us in our information seeking. Search is often regarded as being synonymous with information retrieval but it is not the only option, some of the others being browsing, summarisation, personalisation and recommendation.

Recommendation involves finding items that users might like based on what is understood of their interests. One of the biggest challenges in recommendation is capturing a person's unique characteristics in order to model them better and give better recommendations. It can be difficult to determine if recommendations are optimal when the user can only indicate a degree of success tangentially, which they do by sharing their rating of an item they have experience of. It has been shown that users are willing to interact more with recommenders if it is more transparent and therefore fosters more trust in the results [16]. Such interactivity can be hugely beneficial, so the question that drove us was how can we best capture these characteristics in order to embody both their interests and their current context. The usual interface to recommender systems will list predictions of items which users may be interested in [13], and this offers little incitement to elicit user feedback.

In any given list a user can only rate the items they have experience of, with no opportunity for feedback on unknown items. In addition, a recommendation list can be ambiguous as it is not clear what can be done with it to positively influence the recommendation or even to exert agency within the process. Because of this, while recommendation is ubiquitously part of the online shopping experience it is most frequently seen as an accessory function; users are familiar with the "customers who bought this also bought" panel as the primary manifestation of recommender systems. Ratings and reviews, which play a key part in recommendation are frequently seen as "sharing opinions with other users" rather than "helping the system learn about you". Researchers have provided contemporary re-imaginings of dedicated recommendation systems to better allow people to browse shop items of interest to them, including "conversational" systems that engage users to elicit feedback using methods like asking or proposing items [15].

Item suggestions remain an automated background task that contributes additional information to an otherwise directed task like online shopping. Recent research has taken to exploring methods by which recommendation could be the focus of a system [2], allowing users to more freely exercise their will based on preferences. Methods like critiquing items based on their properties [8] and interactive recommendation [10] have formed the basis for "conversational" approaches which allow for exploration and an active approach to recommendation thus reducing the pressure on eliciting information by making it a primary focus.

These methods of critiquing and interacting are useful in establishing that computer-driven recommendation, with its background in predicting a user's interest *a priori*, can benefit from the direct interaction that happens when people suggest things to each other. Conceptually, if users have a way with which to engage with the system in more ways than just sharing opinions on what has been seen, we have the opportunity to learn more about them. This flexibility results in a much shorter time to produce accurate recommendations [10] and more diverse results [9].

In the work we report here, we explore a new approach to conversation within recommendation as applied to visual item, namely movies. We have developed a way to generate conversation around a large dataset, letting users navigate their recommendations in situations where metadata about items is not present. An application called MovieQuiz, which allows users to quickly navigate the dataset to alter the initial recommendation given to them, is used as a basis for an evaluation of our approach. We recorded user interactions, ratings and responses to a follow-up survey for the purposes of evaluation and we show the ways in which our interactions can improve a user's ability to find good recommendations.

2 Background

Recommendation is traditionally regarded as an information retrieval problem in one of two broad forms [14], collaborative-filtering (CF) and content-based (CB) recommendation, both of which have been studied extensively [4, 13]. CF recommendation attempts to mimic "word of mouth" suggestions, those recommendations users would expect to hear from their friends, by finding people like themselves whose similar tastes can be used to offer likely good items. CB recommendation, by contrast, attempts to classify a user's interests using machine learning and metadata about the items a user likes, in order to find good suggestions.

Content-based recommendation [12] covers any form of recommendation where the algorithm uses intrinsic data from the items it recommends. The drawback of doing this is the requirement that all of the possible items are well-described with metadata before recommendation begins, which comes with the advantage of giving good recommendations from a sparse amount of user data. Case-based recommendation (CBR) [3] attempts to leverage knowledge about prospective items and users interests. In CBR there are a series of known "cases", suggestions that were acceptable in the past, that are then offered to users based on their metadata. CBR is suited to domains such as product suggestion where the items can be described or identified by a well-defined set of metadata such as price, age or colour [18].

Collaborative-filtering [14] (CF) is widely-favoured commercially and offers a number of advantages. It uses only the rating information of items to recommend, either by grouping a user with others who have similar tastes (user-based CF), or uses ratings as a means to describe items having similar rating patterns yielding good recommendations for the people who rated them (item-based CF). With this, as the number of people rating items increases the probability that a match will be found for a new user also increases. By not requiring item metadata for its algorithms, CF is generally useful for recommendation without the need for specialised design for the items it uses.

Recent research has highlighted the need to treat the recommendation process as conversation, an interaction between the user and a system they should trust [19]. In such research, conventional recommendation is paralleled with a conversation, outlining a respectful process that does not place cognitive load on the user by respecting other content it appears with. This shift in approach will highlight that users' rating information provides a better recommendation, rather than being just a mechanism for the user to share opinions with a community. Work on ways to make a conversation between a user and a system possible have centred around case-based recommendation. Leveraging the well-described items in a case-base interaction of the form "I what something like this but less expensive, or a different colour", called critiquing, has been explored [8] with some success, as has "preference-based" [10]. Recent research with case-based conversational recommenders concludes that users prefer a level of control that mirrors their domain knowledge, i.e. someone who knows nothing of cameras will not know what feedback to provide on lens aperture, say [6]. There have also been explorations of recommendation as a game [1] or from a Human Computer Interaction perspective [11].

Here we present a new approach that is influenced by the case-based approach, without depending on an understanding of the domain by either the user or the system. Interaction is through a new type of explicit one-item relevance feedback [7], designed not for search but for CF recommendation.

3 Design and System Outline

Our approach centres around the idea of users choosing their area of interest. We hypothesise that using only the number of ratings and the average rating of items we can reduce the set of items to recommend from in order to offer better recommendations. We provide a means to give feedback based on the reaction, either reasoned or reactionary, of "I don't think I'd like that" or "I'm interested in that". While this reasoning is fussy, imprecise, and difficult to capture it is nonetheless an important part of decision-making for users. In contrast to early work on case-based conversation [10] this is not the same a expressing "I'm interested in more like this", rather the process proceeds like a conversation in which indicating a preference produces potentially entirely new recommendations. Our approach also differentiates a person's *immediate* interests, i.e. in this interactive session's preference indications, from their *continuing* or on-going interests, collected when they rate items.

The strength of CF recommendation lies in using rating information to understand users in comparison to others, to place them in a neighbourhood of peers or find items similar to the ones they like. Our approach uses this understanding of items through ratings, by focusing on how popular an item is, and how well it is rated. The popularity of an item ($Pop(i)$) for our purpose is its rating coverage, i.e. the number of people who have rated it, while the measure of how well rated it is comes from the average rating:

$$\begin{aligned} Pop(i) &= \sum ratings(i) \\ Rated(i) &= Avg(ratings(i)) \\ Point(i) &= (Pop(i), Rated(i)) \end{aligned}$$

From this, any item in the collection can be represented on a graph of popularity against average rating. This graph is a representation of the collection that is equally valid in all areas to user tastes. That is to say that aficionados of items such as books or film can understand there are audiences for both well-rated niche items and items that everyone has seen but wouldn't be their favourite.

Our approach works iteratively which makes it attractive for recommending image-based objects [5]. A session begins with the user having access to the entire collection of items. Two indicative movies are randomly picked from the collection, one to represent popular items and another to represent highly rated ones. The *popular* indicative movie is chosen from the movies with at least half the average number of ratings, while the *highly rated* one is chosen from movies with at least half the average rating of the collection. These are chosen from the movies considered to be *of interest* to the user, the set that they are working to decrease at each iteration. The two options are shown to the user to ask "Which do you prefer?" Additionally, a list of recommendations from the collection is generated and the top five are shown below the question, both to give users a sense that their interaction is having a meaningful effect and to show them new suggestions they may be interested in. Once the user chooses an option, the set of items from which recommendations and interface choices are generated, is partitioned. We use bounding here, which has been explored in search tasks [17] but not in recommendation, especially as a means by which conversation can occur. Here we use lower rather than upper bounds, to signify *least acceptable value.*

A new pair of options, with a list of recommendations, is posed to the user. The degree to which the items are partitioned depends on the density of the collection and our aim is to reduce the set to produce visible change in recommendations through every action. This continues until the user stops answering questions or there are less than ten items to choose from, at which point all ten are presented.

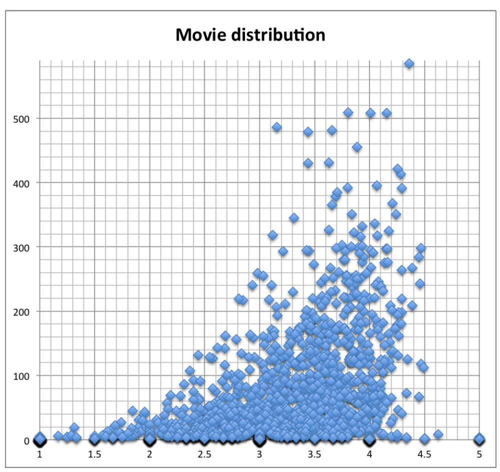

Fig. 1. Distribution of items in the MovieLens dataset plotted using our measurements

We guide the user through a series of decisions that subdivides the recommendation space according to preferences using a pair of lower bounds, reducing the portion of the collection we dub *of-interest* . This differs from critiquing, where the conversation is based on domain-specific traits. Our approach therefore works with items that do not have descriptive metadata, making it useful in situations where none exists.

4 The MovieQuiz Application

We developed an application to evaluate our method using the MovieLens 100K dataset which contains 100,000 ratings from 1,000 users on 1,700 movies. We use this as the seeding data for recommendations, with actual user interaction and rating data collected from other live users. Our example application uses movies, where "blockbuster" films and "indie hits" represent equally valued possible recommendations. Prior to engaging with the conversational interface users were asked to rate 10 of the most-popular films from a presented list.

We use a k-NN item-based collaborative-filtering algorithm to form recommendations. This algorithm is used for traditional recommendation and we adapt it here for our conversational approach as detailed above, to recommend from a subset. The adaptation is conceptually straightforward, in that we modify it to recommend only films with an average rating greater than or equal to X and with Y ratings, where X and Y are determined by the user's interactions with the conversational interface on a per-session basis. Any recommendation algorithm that can be so altered could be used for this approach.

In order to enable traversal of large datasets by the user, the affordance of the interface we develop must allow interaction while informing the user of the current best recommendations. Our basic layout, as shown in Figure 3, is to prompt the user with two candidate preferences. Not shown below the choices is a list of the top five recommended films from the collection according to the current partitioning. Users are given the title and genres of the movie, along with a poster and links to searches for the film on IMDB[1] and YouTube[2].

Experimentally, and as can be seen in Figure 1, the MovieLens dataset shows a skew toward items with higher ratings. This results in users needing to express a preference for high ratings numerous times at the start of a session before any significant changes are seen to their recommendations. For this reason we place greater weight on an interest in films with high ratings at the beginning of the process, incrementing the high rating bound by 2.5 on the first action and 0.5 after that. The popularity bound was incremented by 150 ratings per action, selecting popular over high-rated.

5 Evaluation

5.1 Interaction Analysis

We generated a detailed log for each user to help understand their actions within the system, and to explore the effectiveness of our approach. For any given rating we examined where the user would see that item on a static list of recommendations, to determine if interaction helped the user find the item more easily and what the average prediction error of ratings was, i.e. the degree to which interaction corrected the algorithm's model of the user. We also considered the

[1] http://www.imdb.com
[2] http://www.youtube.com

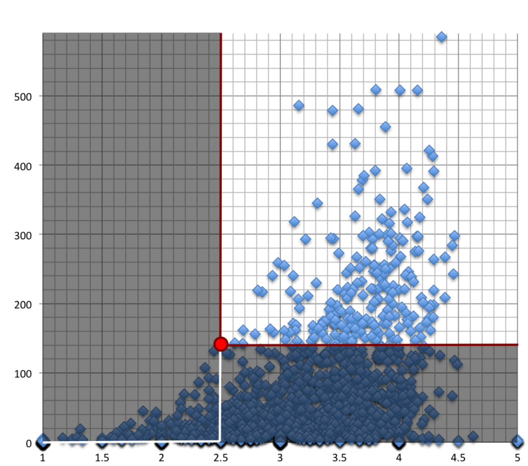

Fig. 2. The collection dissected according to the user's choices in the system

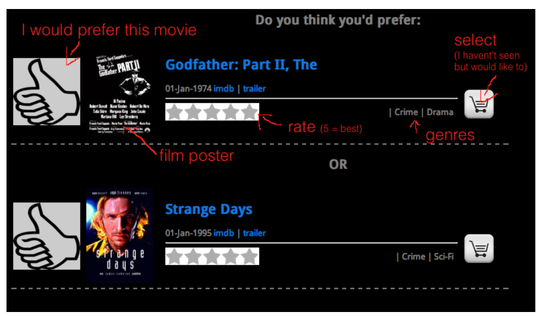

Fig. 3. The MovieQuiz application

average number of *moves* or interactions needed to get to an item that a user rated, a measure of user effort and system efficiency not unlike query difficulty.

We gathered 4,153 intra-conversation ratings from 251 people, and recorded the details of their 2,415 moves within the system. The average number of sessions (complete sets of interactions from start to end) each user had was two, with 9.6 average moves per user. The average user rated 20 items over the course of their sessions, having initially rated 10 items from a non-interactive list before starting (which were excluded from our analysis). Our set of tests involved an examination of where the items that users rated would appear on a flat list of recommendations. In order to test this for each user we used the same item-based collaborative filtering algorithm used in the MovieQuiz application and generated a list of 100 recommendations for them given their initial 10 ratings, made prior to using the interactive interface. Of the 4,153 ratings given while interacting with the system, the recommender algorithm alone lacked sufficient information to recommend 3,704 of the items within the users' top 100. These ratings were therefore excluded from the mean and standard deviation figures generated in Table 1. We also generated figures for the number of moves taken to get to an item worth rating, average rating, and error of predicted rating given by the algorithm.

Table 1. Interaction Analysis

Data	Mean	Std. Dev.
Moves-to-rate	2.33	2.26
Rating	3.60	0.41
Prediction Error	3.27	1.15
List place	77.9	22.3

Our findings, presented in Table 1, show a number of things. If the algorithm recommended an item that the user rated, it was in 78th place on the list on average, with a large deviation. This was the case for only 449 items, the rest being below 100th place on the list. If the recommendations were listed in groups or pages of ten as search results are, then it would take seven actions of "next page" before the user found their item, compared to an average of 2.3 actions in our approach. It follows that our approach would enable users to find the items they were looking for with greater effectiveness. We then looked at how usefully distinct the ratings were and found a reasonable accuracy as defined by RMSE (discussed later), though even so the average prediction error was 3.27, much larger than the RMSE, indicating that the items the user chose to rate were unexpected by the system. These unexpected items could not be accounted for through the algorithm alone and therefore our conversation helped the user find them. The average rating was 3.6 with a standard deviation of 0.4, indicating users expressed opinions on items in a slightly positive way.

Our collaborative-filtering conversation helped users find items that were of interest to them measurably more efficiently than a static recommendation using the same algorithm. We followed this with an exploration of user attitude toward the conversational approach.

5.2 User Survey

After our users had completed their trial use of the system 33 of the 251 users completed a short questionnaire about their prior usage. Of particular interest in the survey was whether users felt that the interaction improved their ability to find good recommendations and whether users without domain specific knowledge, or any knowledge of the items they were asked to judge, were at a disadvantage using the system. Previous research has found that users with greater domain knowledge prefer more fine-grained interaction and conversation from their recommender [6], so we were interested to see if this could be due to other conversational approaches hinging on domain-specific attribute feedback mechanisms such as "Like this but more expensive". The survey included the questions shown below, designed to enquire about users' knowledge levels and their comfort with the system, as a method of finding items and as a series of questions they could answer easily. Questions 1 to 8 were posed using a 5-point Likert scale.

1. How often do you watch movies, either at home or in the cinema?
2. Would you consider yourself knowledgeable about movies?
3. How many of the movies in the system did you recognise?
4. What did you think of the quality of the movies suggested by the system?
5. Did you feel the movie recommender offered a good selection of movies you otherwise wouldn't have heard of/seen?
6. What did you think of the "Which do you prefer" interface?
7. Do you think the interface helped you find good films?
8. How easy was it to state a preference between two movies in the movie quiz?
9. Did you find using the interface preferable to just being given a list of suggestions?
10. Would you use the interface in future, as part of Netflix or Amazon, as a way to help find movies?
11. Any other comments?

We found that users who responded had a wide range of experience and perceived knowledge about movies. The average score for question one, designed to show user experience with the domain area, was 3.27 on a scale of 1-to-5, with a standard deviation of 1.125, showing that while some were experienced, the average had a casual knowledge on the subject. Question two, on the user's own perceived knowledge of film, had an average of 3.33, with standard deviation of 1.163, indicating that for most movies they have at least some knowledge.

Next we looked at users' acceptance of the recommendations generated, finding that responders found the algorithm recommended fair quality films, with one user suggesting a "tag system" be used for genre-specific navigation, i.e. they would like some content-specific features. Users overall felt that the recommender helped them to discover a reasonably diverse set of films they probably wouldn't have seen otherwise, see Figure 4.

Finally, we looked at how users found the interface. Those asked stated they thought the interface was worthwhile with on average an only slightly greater than random chance of recognising films in the system (average score of 3.3, standard deviation of 0.95), suggesting that in a traditional conversational recommender they would have trouble giving feedback on any item features, and preferred a less interactive approach [6]. However with the approach to conversation we used, users felt that it helped them find good items and even without a high degree of domain knowledge they were able to offer feedback (Figure 6). Users preferred our interface to being offered a list of suggestions.

6 Conclusion and Implications

We have shown that it is possible to offer conversation in a recommender system using only rating-derived data, a novel contribution that offsets the more usual reliance on metadata attributes for conversation. While the extent to which

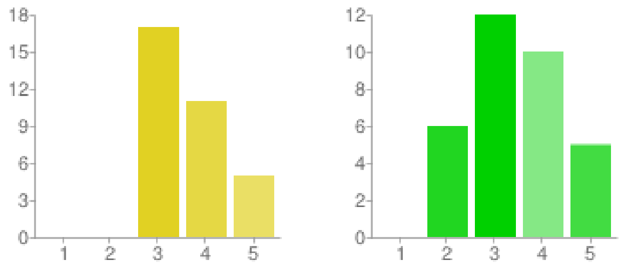

Fig. 4. "What did you think of the quality of the movies suggested by the system", and "did you feel the recommender offered a good selection of movies" ?

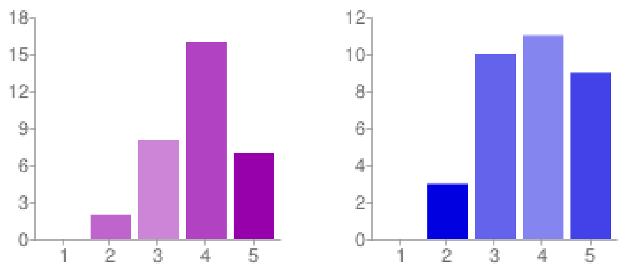

Fig. 5. "What did you think of the interface" and "Do you think the interface help you find good films ?"

users can form information seeking strategies for answering the quiz interface is beyond the scope of this work we have found that users are satisfied with the mechanism we present for responding and finding items without confusion. Also clear is that the explicit information in the form of relative preference statements that can be harvested offer a possible new source of feedback to be harnessed to gain perspective on user information needs.

Finally we explored feedback from users of an application designed to prompt interaction, finding users greatly prefer an interactive interface to being given a list and had no trouble making choices to provide feedback and, in their mind as well as demonstratively, improving their suggestions.

Recent research has said that specific domain knowledge results in a preference for more interaction in recommendation, but here we have shown that a greater degree of interaction need not come with a domain-knowledge barrier, provided it does not hinge on domain specific attributes. Further work could be done to investigate if the variables used to identify an item, popularity and average rating, could be replaced with other valid variables, including possible metadata. Other vectors of investigation possible would include examining whether a hybrid system that limits the items being traversed by metadata, e.g. only films with the genre "action", would produce an improved recommendation.

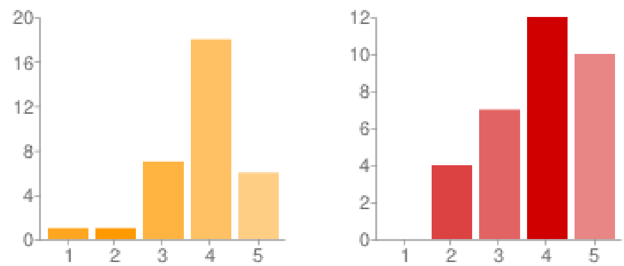

Fig. 6. "How easy was it to state a preference between two movies" and "Did you find using the interface preferable to just being given a list of suggestions ?"

From the user's perspective we offer an entirely new way to receive recommendations, which gives the system a lot of information quickly and transparently. By engaging the user in conversation we improve their ability to find items, in an open way. Given that privacy and the use of personal information are growing concerns in the public eye this transparent approach might also improve user satisfaction with how they are modelled in a recommender system, giving them transparent control of the process of modelling. By designing a conversational method for the least content-rich recommendation approach we have created a method that can in future be incorporated into any recommendation algorithm to allow for interaction without domain knowledge.

Acknowledgements

This work is supported by Science Foundation Ireland under grant numbers 07/CE/I1147 (CLARITY) and SFI/12/RC/2289 (INSIGHT).

References

1. N. Alon, B. Awerbuch, Y. Azar, and B. Patt-Shamir. Tell me who I am: An interactive recommendation system. *Theory of Computing Systems*, 45:261–279, 2009. 10.1007/s00224-008-9100-7.
2. O. Averjanova, F. Ricci, and Q.N. Nguyen. Map-based interaction with a conversational mobile recommender system. In *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM '08. The Second International Conference on*, pages 212–218, Oct. 2008.
3. Derek Bridge, Mehmet H. Göker, Lorraine McGinty, and Barry Smyth. Case-based recommender systems. *Knowledge Engineering Review*, 20(3):315–320, 2005.
4. Ramon López de Mántaras, David McSherry, Derek G. Bridge, David B. Leake, Barry Smyth, Susan Craw, Boi Faltings, Mary Lou Maher, Michael T. Cox, Kenneth D. Forbus, Mark T. Keane, Agnar Aamodt, and Ian D. Watson. Retrieval, reuse, revision and retention in case-based reasoning. *Knowledge Eng. Review*, 20(3):215–240, 2005.

5. Eoin Hurrell, Alan F Smeaton, and Barry Smyth. Interactivity and multimedia in case-based recommendation. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society (FLAIRS) Conference*, pages 347–351. AAAI Press, 2012.
6. Bart P. Knijnenburg, Niels J.M. Reijmer, and Martijn C. Willemsen. Each to his own: how different users call for different interaction methods in recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 141–148, New York, NY, USA, 2011. ACM.
7. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
8. Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. Thinking positively - explanatory feedback for conversational recommender systems. Technical report, In Proceedings of the ECCBR 2004 Workshops, 2004.
9. L. McGinty and B. Smyth. On the role of diversity in conversational recommender systems. *Case-Based Reasoning Research and Development*, 2689:1065–1065, 2003.
10. Lorraine McGinty and Barry Smyth. Evaluating preference-based feedback in recommender systems. In Michael O'Neill, Richard Sutcliffe, Conor Ryan, Malachy Eaton, and Niall Griffith, editors, *Artificial Intelligence and Cognitive Science*, volume 2464 of *Lecture Notes in Computer Science*, pages 209–214. Springer Berlin / Heidelberg, 2002.
11. Sean M. McNee, John Riedl, and Joseph A. Konstan. Making recommendations better: an analytic model for human-recommender interaction. In *CHI '06 extended abstracts on Human factors in computing systems*, CHI EA '06, pages 1103–1108, New York, NY, USA, 2006. ACM.
12. Michael J. Pazzani and Daniel Billsus. Content-based recommendation systems. In *The Adaptive Web*, pages 325–341, 2007.
13. P. Resnick and H.R. Varian. Recommender systems. *Communications of the ACM*, 40(3):58, 1997.
14. Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
15. H. Shimazu. Expertclerk: navigating shoppers' buying process with the combination of asking and proposing. *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, pages 1443–1448, 2001.
16. Swearingen Sinha, Kirsten Swearingen Rashmi, and Rashmi Sinha. Beyond algorithms: An HCI perspective on recommender systems, 2001.
17. Alan F. Smeaton and C. J. van Rijsbergen. The nearest neighbour problem in information retrieval: An algorithm using upperbounds. In *Proceedings of the Fourth SIGIR Conference*, pages 83–87. ACM, 1981.
18. Barry Smyth. Case-based recommendation. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 342–376. Springer Berlin / Heidelberg, 2007.
19. Daniel Tunkelang. Recommendations as a conversation with the user. In *ACM RecSys*, pages 11–12, 2011.