Chapter 8
Evaluating Recommender Systems for Supportive Technologies

Jill Freyne and Shlomo Berkovsky

Abstract Recommender systems have evolved in recent years into sophisticated support tools that assist users in dealing with the decisions faced in everyday life. Recommender systems were designed to be invaluable in situations, where a large number of options are available, such as deciding what to watch on television, what information to access online, what to purchase in a supermarket, or what to eat. Recommender system evaluations are carried out typically during the design phase of recommender systems to understand the suitability of approaches to the recommendation process, in the usability phase to gain insight into interfacing and user acceptance, and in live user studies to judge the uptake of recommendations generated and impact of the recommender system. In this chapter, we present a detailed overview of evaluation techniques for recommender systems covering a variety of tried and tested methods and metrics. We illustrate their use by presenting a case study that investigates the applicability of a suite of recommender algorithms in a recipe recommender system aimed to assist individuals in planning their daily food intake. The study details an offline evaluation, which compares algorithms, such as collaborative, content-based, and hybrid methods, using multiple performance metrics, to determine the best candidate algorithm for a recipe recommender application.

J. Freyne (✉)
Information Engineering Laboratory, ICT Center, CSIRO, GPO Box 76, Epping, NSW 1710, Australia
e-mail: jill.freyne@csiro.au

S. Berkovsky
Information Engineering Laboratory, ICT Center, CSIRO, GPO Box 76, Epping, NSW 1710, Australia

Network Research Group, NICTA, Locked Bag 9013, Alexandria, NSW 1435, Australia
e-mail: shlomo.berkovsky@csiro.au; shlomo.berkovsky@nicta.com.au

E. Martín et al. (eds.), *User Modeling and Adaptation for Daily Routines: Providing Assistance to People with Special Needs*, Human–Computer Interaction Series,
DOI 10.1007/978-1-4471-4778-7_8, © Springer-Verlag London 2013

8.1 Introduction

Recommender systems have evolved in recent years into sophisticated support tools that assist users in dealing with the decisions encountered in everyday life. Recommender systems were designed to be invaluable tools in decision making situations, where large number of options are available, such as deciding what to watch on television, what information to access online, as well as what to purchase in a supermarket, and what to eat. Recommender systems can play an increasingly valuable role when considered in the context of users with special needs, as illustrated through this chapter.

Evaluating the accuracy and effectiveness of recommender systems is a challenge, which has been faced since their inception [31]. Evaluations are carried out during the design phase of recommender systems to understand the suitability of approaches to the recommendation process, in the usability phase to gain insight into interfacing and user acceptance, and in live user studies to judge the uptake of recommendations and impact of the recommender system.

In this chapter, we present a detailed overview of evaluation techniques for recommender systems covering a variety of methods and metrics suitable for this task. We detail the three typical evaluation paradigms for recommender systems – offline analysis, user studies, and online studies – and provide examples of each paradigm in the context of every day activities and people with special needs. We also detail evaluation metrics suitable for judging algorithm performance in terms of accuracy and important usability dimensions. We highlight the use of various study types and metrics by presenting an evaluation case study of a recommender system for people with special dietary requirements. The evaluation focusses on a meal recommender application for assisting users in planning their daily food intake. The study details an offline evaluation, which compares a set of recommender algorithms, collaborative, content-based, and hybrid algorithms, using multiple performance metrics. Also included is a discussion around suggested scenarios for other evaluation paradigms, including a usability study and a live online evaluation.

The chapter is structured as follows. In Sect. 8.2, we summarise the traditional recommender system evaluation paradigms and metrics. Section 8.3 details a large scale evaluation conducted with an interactive meal planning application and discusses alternative evaluation scenarios and research questions that can be investigated. Finally, Sect. 8.4 concludes the chapter.

8.2 Evaluation Techniques for Adaptive Recommender Systems

Evaluation of the performance of recommender systems generally follows one or more of three paradigms [15, 33]. *Offline evaluations* seek to learn from data that has already been gathered and typically take the form of simulated experiments.

The outcomes facilitate the tweaking and polish of algorithms and processes used to generate recommendations. *User studies* typically involve a small cohort of participants undertaking specific tasks on a prototype system and outcomes include feedback on a variety of areas, including interfaces, algorithm performance, and general system acceptance. *Online evaluations* learn by monitoring real users "in the wild", as they interact with a live system and the outcomes include real-time, longitudinal data that facilitates understanding of algorithm performance and system appeal. Each paradigm and analysis provides researchers with different information pertaining to a multitude of possible system components, from algorithm performance, through preferred user interaction mechanisms, to real-world impact. While many systems employ one or, perhaps, two types of analysis, there is a logical evolution from offline evaluation, through user studies, to online analysis.

8.2.1 Offline Experiments

Much of the early work in the recommender systems domain focused on offline algorithm accuracy evaluations and preference predictions in a variety of application domains, e.g., movies, restaurants, television, and books. For most recommender system designers, offline algorithmic analysis is the first crucial step in designing an adaptive system with recommendation support. There are a large number of tried and tested recommender system algorithms and approaches, which are well documented in the literature. These include content-based algorithms [20], social or collaborative algorithms [18, 19], and complex machine learning algorithms [28, 35] and there are many variations of each. This phase of evaluation is primarily conducted offline with datasets collected for the purpose of simulated experiments, such that a high degree of control remains with the researcher as to what is analysed.

Offline analyses typically focus on the predictive power of approaches and algorithms in accurately determining the opinions of users [7, 16]. This is achieved using simulated user interactions, such as providing ratings or selecting items from an item repository. Often, portions of user profiles are withheld from the algorithms, with ratings being predicted and the predictions then compared to the real ratings. The advantages of offline experiments include the provision for a large selection of algorithms to be evaluated at low cost and without the requirement for real-time user input. Offline analyses facilitate thorough investigation of various components of algorithmic performance, including coverage, accuracy, execution speed, the susceptibility to issues such as the cold start problem, and many other dimensions which impact directly on algorithm performance and are difficult to evaluate in deployed systems.

The quality and applicability of the knowledge gained from offline experiments is often highly correlated with the quality, volume, and closeness of the evaluation dataset to the data which would be collected by the intended recommender system. This is a key consideration for offline experiments. If the data gathered comes from users, who are not typical of the intended audience, if the items do not have

the same features, or if the system has different functionality or context, then the lessons learned are less clear. There exists several publicly available datasets for recommender system evaluation; the most widely used are the MovieLens [14], EachMovie [22], and Moviepilot [32] movie rating datasets. While the availability of these datasets has proven invaluable in the development of recommender systems algorithms, their use in the design of adaptive systems similar to those discussed in this book is often limited due to a miss match in domain and recommendation functionality.

There are numerous examples of offline evaluations in recommender systems [3, 6, 15, 23]. Burke's investigation into hybrid recommender systems is a typical example of an offline analysis comparing multiple recommender systems algorithms [6]. The aim of the analysis was to judge how effective each algorithm (collaborative filtering, collaborative heuristic, content-based and knowledge-based) is at recommending restaurants to users of the Entree restaurant recommender system. As the evaluation did not call for exact rating predictions, rank accuracy metrics (see Sect. 8.3.2.3) were used to compare the algorithms. The data set was collected from the Entree system itself and user ratings were extracted from the system logs. The obtained results showed that the performance of the algorithms varied, but that the collaborative algorithms generally performed best. Burke used these findings in further analyses of the performance of hybrid recommender algorithms.

In the domain of daily routines, a detailed analysis of recommender systems in television program scheduling can be found in the Neptuny's ContentWise recommender system [2]. With more and more digital entertainment options available at the touch of a button, the experience of watching television has changed drastically in recent years. Internet Protocol Television (IPTV) delivers traditional TV channels and on demand TV over broadband networks, meaning that users can draw from a huge repository of programs and create their own schedules and playlists. The challenge for providers is that viewing becomes interactive and there are a range of opportunities and challenges for personalization and recommender technologies to assist users in finding and engaging with relevant content [2]. The ContentWise recommender system was integrated into a live IPTV provider's service and the data gathered through the live site has facilitated analyses, which involved three versions of the ContentWise recommender system: item-based collaborative filtering, LSA content-based algorithm, and collaborative SVD algorithm. The data used in the analyses was based on the views recorded during 7 months of user activity from a video on demand catalogue. The analyses concentrated on evaluating the predictive accuracy metrics using recall (see Sect. 8.3.2.3). The results showed differences in the performance across the algorithms, with the collaborative algorithm outperforming other algorithms.

8.2.2 User Studies

While investigating which techniques and algorithms work best in certain domains, their accuracy and predictive power is only one of many measurable components

that contribute to the success and impact of the adaptive system [34]. Previously gathered information can provide insight into user behaviour patterns, but it is often difficult to accurately simulate how users will interact with a system and even more challenging to effectively judge the real-world impact of a system. Many researchers have argued that the predictive accuracy of a recommendation algorithm might not correlate with the user perceived value of the recommendations or the general appeal of the service or system, which is often impacted by the visual and interaction design, language, tone and general usability [5, 8]. Thus, researchers frequently turn to user studies to observe interactions of real users with the systems in order to obtain real-time feedback on performance and perceived value.

User studies typically involve the recruitment of a small cohort of users to complete specific tasks and provide feedback on a prototype system [21,27]. User studies can gather qualitative and quantitative feedback on the system performance, often logging each and every interaction, monitoring task durations, completion rates, as well as gathering explicit feedback on interface, performance, and preferences relating to user experience. In systems, where change or awareness is sought, users are often requested to fill out questionnaires before, during, and after exposure to a system or technology. This user feedback can be used to confirm researcher's hypotheses and inform changes in the service design and interaction methods. For example, they can determine the most appropriate layout of a recommendation engine in a larger system or the type of rating scale that users find intuitive. More importantly, researchers can acquire real-time feedback on various aspects and functionality of the service provided by the system.

In the area of recommender systems for information access, the ASSIST social search and navigation system was evaluated in a classroom-based usability study [10]. ASSIST was designed to recommend Web search results and navigation paths within a repository of research papers by exploiting recommender algorithms and visual cues. The purpose of the study was to assess the actual and perceived value of social support in search, and the integration of social search and social browsing. The study gathered quantitative and qualitative feedback from participants. Two versions of the system were created: a control system that had no recommendation functionality and an experimental system that provided users with a host of social support features. Thirty students were recruited and randomly assigned to the experimental groups. The students were asked to spend 1 hour using the system in order to locate papers pertaining to the introduced topics and provide a short explanation justifying the relevance of each, before filling out a questionnaire relaying their experience with the system and their views of the various features provided. The evaluation examined the output quantity and quality, as well as rank accuracy metrics of the recommendations, while also facilitated an understanding of real user interactions, the impact of visual cues, and the critique from the students involved. Results showed that users found more relevant results when supported by the recommender system but feedback on the visual cues reported that there were too many cue types which were not intuitive to all users.

Pixteren et al. [26] in their work on intelligent meal planning assistance, modelled the similarity of recipes by extracting important features from the recipe

text. Based on these features, a weighted similarity measure between recipes was determined and this provided the foundation for their recipe recommender engine. In order to judge the accuracy of the models, they conducted a user study, in which real users were asked to provide their opinion on the similarity of recipes that was then compared to various predictive models. Over a period of 2 weeks, 137 participants were recruited through emails and message boards. Participants were presented with 20 consecutive recipe pairs and for every pair they were asked to rate the similarity on a 7-point scale. The recipe presentation interface showed the title, cuisine, preparation time, ingredients, and directions. The recipe similarity measure derived by the authors was compared to that of a baseline similarity metric, cosine similarity, and the users' explicit similarity score. Results showed that the accuracy of the derived similarity metric outperformed that of the baseline algorithm.

The opportunity for diverse and detailed feedback through user studies is of immense value. However, user studies come at a heavy cost in terms of user time and (if the participants are not volunteers) potential financial costs, which can limit the number of system dimensions being investigated. Revisiting the recipe recommender example of [26], we note that the evaluation mainly concentrated on model accuracy and ignored other dimensions, such as algorithm accuracy. In a similar vein to the offline experiments, care must be taken when recruiting test subjects of user studies, to ensure that they represent the intended audience of the resulting live system.

8.2.3 Online Evaluations

The most realistic assessment of a recommender system can be achieved by an online evaluation or live user study. This typically involves a group of trial users, who use the system in true information overload conditions and are assisted by the system in performing self selected tasks. It should be noted that live online evaluations generally follow a number of offline and/or user studies, or are exploited in situations, where the performance can more accurately be measured in real-world scenarios, such as with systems that influence long-term user behaviour [13]. Only an online study with real users, who are self motivated to try a system and use it in a natural manner, can enable researchers to monitor the true system impact in its intended environment [17]. In addition, research which applies recommendation technology to new application domains or populations where datasets do not exist, or to complex environments that cannot be simulated, also require online experimentation [11].

In online studies, users are often exposed to various instantiations of a system, which may focus on different algorithms, interfaces, or other variables. While several dimensions of a system can be experimented upon, typically most variables are kept stable and only the one being investigated is adapted. To evaluate the dependent variable, user interactions with the system are monitored over a period of time and then analysed. For example, the uptake or rejection of recommendations,

the ranking or position of the selected items in the recommendation lists, and the resulting user behaviour can be examined to determine the outcomes of the evaluation. The recruitment of users for online evaluation can be on a voluntary basis, through random selection from an existing user base on a pre-existing site, or all system users can be involved in a trial.

Vigourous research into recommender system performance in online studies has been carried out by the GroupLens research group on the live MovieLens recommender system [18]. The research platform, which was established in 1997 and now has over 100,000 users is an ideal platform for small and large scale live evaluations in movie recommendations. The MovieLens team has carried out multiple user evaluations, a number of which have looked at the practicality of obtaining accurate user models with minimal user effort and the impact of this data in the recommendation process [29, 30]. Ideally, systems need to elicit high-value user information that holds important knowledge about user preferences in its early interactions with users. For example, acquiring a rating for movie that received mixed reviews, which acts as an informative differentiator in a dataset is more valuable than acquiring a positive rating for a movie that is liked by most users. Movielens researchers devised several strategies to select movies that new MovieLens users should rate before they receive recommendations. An online study was conducted for 20 days and 381 users were involved. To assess their efforts, each user was randomly assigned to a condition and asked to rate 15 movies in order to complete their registration. The movies presented for rating were determined by varying algorithms. In line with previous offline analyses, two algorithms showed performance benefits for users, in that users were shown fewer titles of movies before they found the 15 that they could rate. However, the data gathered by a lesser preforming algorithm (in terms of selecting movies the user could rate) led to the generation of more accurate recommendations and users did not perceive rating the additional movies as effort. Thus, it was deemed the best suited algorithm for this environment. Without the completion of the live analysis in this case, the authors may have misplaced emphasis on user effort and possibly compromised the performance of the final system.

The analysis of the ContentWise system for IPTV video on demand recommendations, discussed in Sect. 8.2.2, continued with an online evaluation that examined the success of each algorithm on the live site. In the online user evaluation, the authors concentrated on responses of real users to recommendations provided by the system and on the recall of these algorithms, as measured by the uptake of recommendations. The impact of the presence of recommendation technology on the system was also measured, as reflected by the number of recommended movies that have actually been viewed within a certain time period after being recommended. Authors monitored user interactions with the system for 24 h and 7 days from the recommendation delivery. Results showed a 24% success rate over the 7 days, but noted differences in success rates between popular and unpopular content, with higher success rates achieved when less popular content was recommended. This could be caused by either the fact that a user has already watched a popular recommended movie or was not interested in the movie at all.

Results also showed a 15% increase in viewing rates associated with the presence of the recommender system. This type of information can only be ascertained through a live analysis.

8.3 Case Study: Analysis of Recommender Algorithms to Support Healthy Eating

To illustrate the considerations and practicalities involved in evaluating a recommender system for daily routines, in particular in the context of supporting dietary choices, one of the most common daily routines we present the following case study. We open our discussion by motivating the urgent need for digital tools supporting users in fighting obesity, before presenting a large scale offline study that provides an understanding of the applicability of several recommender algorithms for the purposes of recipe recommendation. We detail the challenges surrounding data collection, algorithm selection, evaluation metrics, and the obtained results. Finally, we discuss future studies demonstrating the user study and online paradigms, which would compliment the lessons learned from the offline evaluation.

8.3.1 Obesity and Daily Routines

Food and diet are complex domains for adaptive technologies, but the need for systems that assist users in embarking on and engaging with healthy living programs has never been more real. With the obesity epidemic reaching new levels, many practitioners are looking for novel and effective ways to engage users and sustain their engagement with online solutions for effective change in everyday life.

A huge challenge facing dieters is to break habits around exercise and food consumption, in order to balance energy intake and expenditure levels. This can be a daunting task, which is often circumvented by dietary providers supplying one size fits all meal plans. While this might be a short-term solution, it is not conducive to long-term behaviour changes due to two primary factors: (1) specified plans are often restrictive and may be too difficult or repetitive for dieters to maintain, and (2) users may not acquire diet management skills that influence the long-term success of the dietary change. On the flip side, asking users to plan from scratch is often equally daunting, given the range of existing food options and combinations available to them.

With the move to digital recording through online or mobile applications, diet solution providers have access to rich records, which encapsulate user preferences for foods and recipes and offer rich input for adaptive support systems. The goal of the presented study was to design and evaluate an adaptive meal planner application

that exploits this rich digital record acquired to assist users in planning meals which not only conforms to the preferences of individual dieters in terms of the foods they like, their cooking skills, budget, and other parameters but also to the rules and guidelines of a particular diet. This tool can assist dieters in acquiring the necessary skills and communicate the implications of certain dietary choices through real-time visual feedback.

8.3.2 Recommender Strategies for Dietary Planning

The domain of food is varied and complex and presents many challenges to the personalization research community. To begin with, thousands of food items exist, so the initial range of choices is immense. Secondly, food items are rarely eaten in isolation, with a more common consumption tending to be in combination, in the form of meals. Given the number of food items, the number of resulting combinations is exponentially large. More complexly, users' opinion on ingredients can vary quite significantly based on several factors. Specifically, the content or ingredients of a meal is only one component that can impact a user's preference. Other components include the cooking method, ingredient availability, complexity of cooking, preparation time, nutritional values, and ingredient combination effects. Finally, cultural and social factors are often crucial in eating and cooking considerations. Add to this the sheer number of ingredients, the fact that eating often occurs in groups, and that sequencing is important, and the complexity of challenge becomes evident.

Recommender systems offer a promising means to address this challenge. They can simplify the task of selecting and planning meals and provide recommendations for meals that both satisfy diet requirements and comply with user preferences. Most traditional recommendation algorithms can be exploited for meal recommendation purposes. For example, a content-based recommender could exploit user preferences for specific ingredients or cooking methods and select meals that include these, whereas a collaborative recommender would find people with similar culinary tastes and select meals they liked. Likewise, a variety of hybrid solutions can be implemented and deployed by a meal recommender.

Figure 8.1 shows a sample user interface to illustrate the recipe recommender. The individual's daily plan is shown in the centre, a structured tree of recipes is on the left, and the recommended recipes are on the right. Users can drag-and-drop their preferred recipes to/from the daily plan and the recommended list changes accordingly. The key to maintaining a diet is often not in the appropriateness of individual meals or dishes to the diet, but in the appropriateness of the combination of meals included in a daily plan. Hence, items in the recommended list are filtered by their compliance with the daily dietary guidelines and the user's current plan, i.e., only items, which would keep users compliant with the diet plan for the day, are shown in the recommendation list. Hence, it is important for the recommender

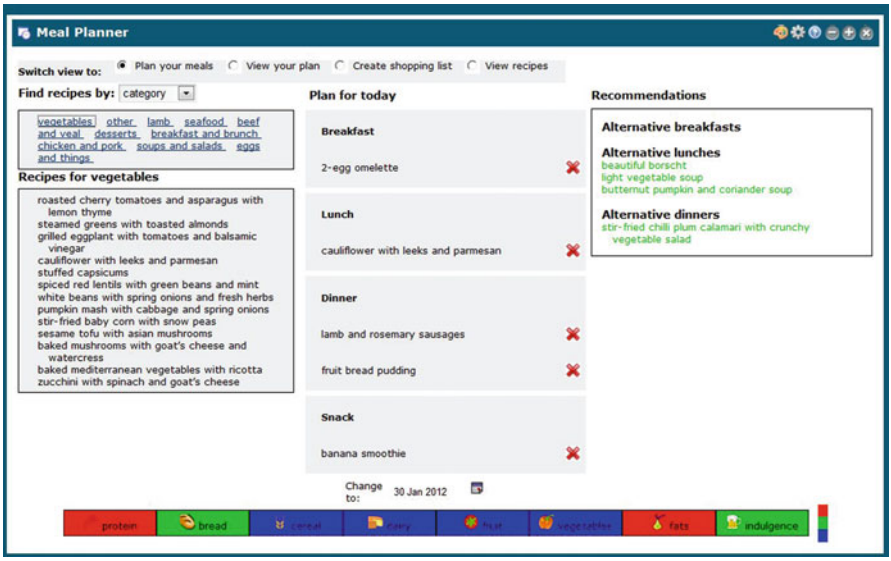

Fig. 8.1 Recipe recommender interface

system to not only be aware of the individuals preferences, but also of the context in which the recommendations are being delivered, so that it can adapt appropriately.

8.3.2.1 The Data

To gather ratings required for the offline analysis, users of Amazon's Mechanical Turk[1] crowdsourcing tool were requested to provided explicit preference information on a set of recipes. Online surveys, each containing 35 randomly selected recipes were posted and users could answer as many surveys as they chose. Users were asked to report on how much each recipe appealed to them on a 5-Likert scale. A set of 343 recipes from the Total Wellbeing Diet recipe books [24, 25] and an online meal planning service Mealopedia[2] was acquired. Each recipe had a common structure, containing a *title*, *ingredient list*, and *cooking instructions*. Two indicators of recipe complexity (the *number of ingredients* in a recipe and the *number of steps* required to cook it) were automatically extracted from this information. We collected 101,557 ratings of 917 users, such that each user rated on average 109 recipes, with the minimum number of ratings per user being 35 and the maximum being 336. The distribution of recipe rating scores over the entire set of users is shown in Table 8.1.

[1] http://www.mturk.com

[2] http://www.mealopedia.com

Table 8.1 Distribution of user ratings

Not at all	Not really	Neutral	A little	A lot
15%	15%	20%	25%	25%

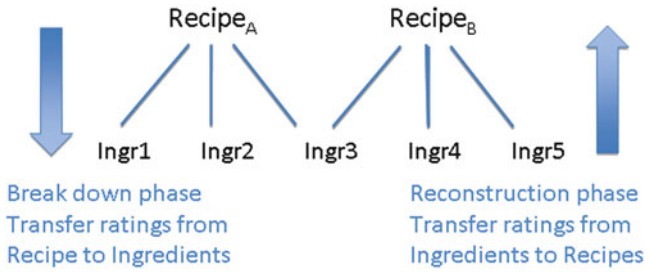

Fig. 8.2 Recipe – ingredient breakdown and reconstruction

8.3.2.2 The Algorithms

In the study, we analysed the applicability of four personalized recommender algorithms: content-based filtering, collaborative filtering, and two hybrid approaches.

A classic example of a cognitive personalization is a *content-based* (CB) recommender system, whose user models represent the degree to which various domain features (in this case ingredients) are liked by the user. CB recommenders promote items, whose features match the features that are preferred by the user [1, 4, 20, 20]. CB filtering examines the ingredients of the recipes and the user's preferences for these ingredients in order to predict the probability that the user will like other recipes in the dataset.

The dataset in this case study represented user preference information on recipes rather than individual ingredients. A number of approaches to working with this data were considered. For example, it would be possible to try identify the effect of recipe complexity and obtain alternative preferences for ingredients through use of a logistic regression. Similarly, general population levels of ingredients could be obtained and exploited. However, previous research showed that simple conversions from recipe ratings to ingredient ratings provided sufficiently good accuracy levels, such that in this study the same method of conversion of recipe preferences into ingredient preferences was used [12]. This content based recommendation process firstly converts the rating for a recipe r_i provided by user u_a to ingredient scores, as schematically shown in Fig. 8.2. The pre-processing step assigns the ratings provided by u_a to each ingredient according to Eq. 8.1. All cooking processes and combination effects are ignored and all ingredients are considered to be equally important. Ratings gathered on recipes are transferred equally to all ingredients, and vice versa, from ingredients to their associated recipes. Once completed, a

content-based algorithm shown in Eq. 8.2 is applied to predict a score for the target recipe r_t based on the average of all the scores provided by u_a on ingredients $ingr_1, \ldots, ingr_j$ making up r_t.

$$score(u_a, ingredient_i) = \frac{\sum_{l \ s.t. \ ingr_i \in r_l} rat(u_a, r_l)}{l} \tag{8.1}$$

$$pred(u_a, r_t) = \frac{\sum_{j \in r_t} score(u_a, ingr_j)}{j} \tag{8.2}$$

Collaborative filtering (CF) algorithms exploit statistical techniques to identify common behavioural patterns amongst a community of users [1, 9, 18, 19]. The recommendations are based on the notion that users who agreed in the past, are likely to agree again in the future. Thus, CF uses the opinions of like-minded users to predict the opinion of a target user. User opinions can be either expressed explicitly on a predefined scale of values or inferred implicitly from the observed user activities. The main stages of CF are to recognise commonalities between users and compute their similarity; select a set of most similar users referred to as neighbours; and aggregate the opinions of the neighbours to generate recommendations. A key advantage of CF algorithms is that they are domain agnostic and require no knowledge of domain features and their relationships. We implemented a standard CF algorithm that assigns predicted scores to recipes based on the ratings of a set of N neighbours. Briefly, N neighbours are identified using Pearson's correlation similarity measure shown in Eq. 8.3 where the similarity of users u_a and u_b is determined by considering the scores provided by each user for the set of items, I_{ab} rated by both u_a and u_b. The prediction for recipe r_t not previously rated by the target user u_a is generated using Eq. 8.4 which considers the ratings provided by N weighted by their similarity to u_a as in Eq. 8.4.

$$sim(u_a, u_b) = \frac{\sum_{i \in I_{ab}} (u_{a_i} - \overline{u_a})(u_{b_i} - \overline{u_b})}{\sqrt{\sum_{i \in I_{ab}} (u_{a_i} - \overline{u_a})^2} \sqrt{\sum_{i \in I_{ab}} (u_{b_i} - \overline{u_b})^2}} \tag{8.3}$$

$$pred(u_a, r_t) = \frac{\sum_{n \in N} sim(u_a, u_n) rat(u_n, r_t)}{\sum_{n \in N} sim(u_a, u_n)} \tag{8.4}$$

Two *hybrid* strategies that combine CB and CF recommendation techniques were also implemented. These break down each recipe rated by u_a into ingredients and exploit CF to reduce the sparsity of the ingredient scores by predicting scores for ingredients with no available information. The $hybrid_{recipe}$ strategy identifies a set of neighbours based on ratings provided on recipes as in Eq. 8.3 and predicts scores for unrated ingredients using Eq. 8.4 (applied to ingredients scores rather than recipe ratings). The $hybrid_{ingr}$ strategy differs only in its neighbour selection step: user similarity is based on the ingredients scores obtained after the recipe break down rather than directly on the recipe ratings. In both cases, the CB prediction shown in

Eq. 8.2 is used to generate a prediction for r_t using the denser ingredient score data. In addition, we implemented a baseline algorithm *random* that assigns a randomly generated prediction score to a recipe.

8.3.2.3 The Metrics

There is a plethora of approaches appropriate for evaluating the performance of recommender systems. The decision on which approach or combinations of approaches to use is informed by the goals and settings of the evaluation. The work of Herlocker et al. [15] and Shani and Gunawardana [33] set out classifications for recommender system performance measurements. Two primary categories of evaluation metrics are suggested to compare the accuracy of different recommender algorithms: predictive accuracy metrics and classification accuracy metrics.

Predictive accuracy metrics show how close a recommender system's predictions are to real ratings given by users. These are deemed to be particularly important in illustrating to users through visual cues the predicted values of items or in ranking items according to their relevance. This category includes the well known and commonly used metric of Mean Absolute Error (MAE) and similar metrics, such as Normalised Mean Absolute Error (NMAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE).

MAE measures the absolute deviation between a predicted rating, $pred(item_i)$, and the true rating, $rat(item_i)$ for user u_x and item $item_i$ as shown in Eq. 8.5.

$$MAE = \frac{\sum_{i \in I} \mid pred(item_i) - rat(item_i) \mid}{I} \tag{8.5}$$

MAE is seen as the standard accuracy prediction metric, as it quantifies prediction errors, is easy to comprehend, and has well studied statistical properties that allow significance testing to be easily computed. Other metrics in this category often appear in addition to or as a substitute for MAE. For example, NMAE normalises the MAE values with respect to the range of ratings and allows direct comparisons across datasets, whereas MSE and RMSE square the error before averaging it to penalise large prediction errors.

Classification accuracy metrics measure the frequency with which a recommender system makes correct and incorrect decisions about whether an item is relevant or irrelevant. These metrics do not predict actual ratings, but concentrate on classifying items into the relevant/irrelevant category. The key to using these metrics is that the user preference information must be represented in a binary relevance form, although this is often too coarse-grain in recommender systems. In order to compute the classification accuracy metrics, the typical 5 or 7 point rating scale is reduced into the binary relevance indicator. Decisions regarding the cut off point are often subjective and depend on the system functionality. Furthermore, different

users may have their own cut off points; some may class a 3 and above on a 5 point rating scale as positive, whereas others 4 and above.

Popular metrics in this category include Precision, Recall and F-measure (FI), which were originally used in information retrieval systems, but have been successfully adopted in recommender systems. Precision measures the proportion of relevant recommendations among all the recommendations (Eq. 8.6). Recall measures the proportion of relevant recommendations among all potentially recommendable items (Eq. 8.7). In many cases, knowledge of both precision and recall is required to effectively judge performance. Thus, both measures can be combined into the F1 metric, which represents their harmonic mean assigning equal weights to precision and recall is shown in Eq. 8.8.

$$precision = \frac{I_{rs}}{I_s} \tag{8.6}$$

$$recall = \frac{I_{rs}}{I_r} \tag{8.7}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{8.8}$$

Two other evaluation metrics should be mentioned. Rank accuracy reflects an algorithm's ability to produce a list of recommended items, ordered according to the user's preferences. Although rank accuracy metrics are more sensitive than classification accuracy metrics in that they order all items in terms of their predicted preference, they are not intended to judge predicted rating accuracy but just the relative relevance of items to an individual. Coverage reflects an algorithm's ability to generate recommendations regardless of their accuracy. It is computed by considering the proportion of items or users for which the algorithm can generate any prediction. Item space coverage refers to the percentage of items, for which a recommender can make recommendations, and user space coverage refers to the percentage of users for which the algorithm can make recommendations.

8.3.3 Offline Evaluation

The metrics selected in the case study evaluation were informed by the nature of the data gathered (more than 100,000 ratings on a set of recipes) and the intended type of analysis (offline evaluation of the applicability of several recommendation algorithms for recipe recommendations). A leave-one-out strategy was employed for the majority of the experiments. For each iteration, one $\{u_i, r_t, rat(u_i, r_t)\}$ tuple was withheld from the data and the algorithms were applied to predict the rating $rat(u_i, r_t)$. A set of 20 CF neighbours was selected and their ratings were aggregated in a weighted manner. Cut off points of 3 and 4 were used as classification accuracy relevance indicators.

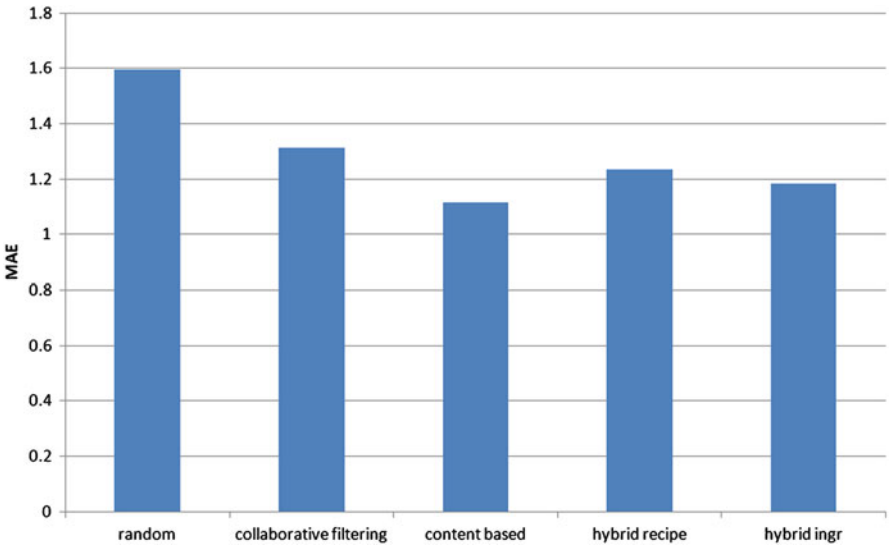

Fig. 8.3 MAE scores

The MAE [15], precision, recall, F1, and coverage scores obtained by each algorithm and averaged across the entire population of users is presented. Figure 8.3 shows the average MAE of the predictions for each algorithm presented in Sect. 8.3.2. The accuracy of the CF and CB recommenders is similar, with an improvement in accuracy of 0.05 over CF obtained by CB. A comparison between the CF algorithm, which treats each recipe as one entity and ignores its ingredients, and the CB algorithm, which considers the ingredients, shows that even the uniformly weighted break down and reconstruction offer improvement in accuracy. The two hybrid strategies $hybrid_{recipe}$ and $hybrid_{ingr}$ produce MAE scores of 1.23 and 1.18, respectively. Hence, the $hybrid_{ingr}$ algorithm is the best performing of the traditional recommender algorithms.

Given the context of recipe recommendations, the prediction accuracy isn't the only suitable metric. In this recommender, the aim is to assist users in planning healthy and appealing meal plans, but the plans will contain a variety of meals over a period of time. Thus, the system should be able to identify a set of appealing meals rather than just the single most appealing meal. In line with this, classification accuracy analyses were carried out to judge each algorithm's ability to produce an accurate positive and negative classification of recipes.

Figure 8.4 shows the classification accuracy of the algorithms. This shows what portion of the predicted scores is converted into correct binary relevance indicators, regardless of their relevant/irrelevant value. Figure 8.4 shows the performance of the algorithms with two cut off points: a strict one that categorises meals with predicted scores of 4 and higher as relevant and a lenient one that categorises meals with

%endfigure

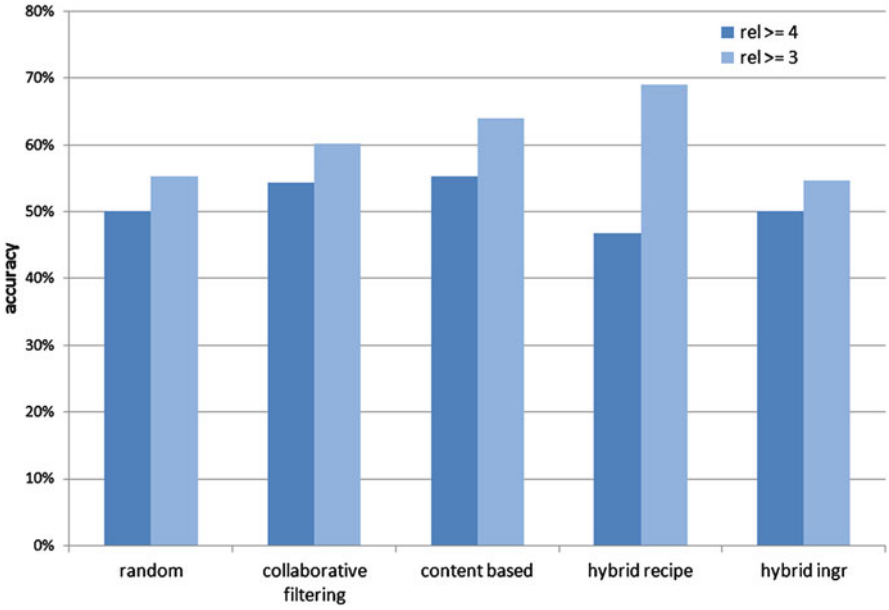

Fig. 8.4 Classification accuracy

scores of 3 and higher as relevant. Different trends were observed, depending on the cut off points. When the strict cut off was in place, the best performing algorithm, with a classification accuracy of 55% was CB, whereas when the lenient cut off was in place, the $hybrid_{recipe}$ algorithm achieved the highest accuracy at 70%.

For a full comprehension of the algorithms' potential, the overall classification accuracy is not sufficient, as only the relevant predictions will determine the recommendations delivered to the user. Thus, *recall*, *precision*, and *F1* measures were calculated for the top 1, 3, 5, and 10 predictions generated by each algorithm. Figure 8.5 shows the average precision of the algorithms when k (1, 3, 5, and 10) recommendations are generated for each user in the dataset.

High precision of the CB and CF algorithms and low performance of the hybrid methods were observed. Figure 8.6 shows the average recall[3] of the algorithms for the same values of k. We observe the CB and CF algorithms again outperforming the hybrid approaches. Combining precision and recall into F1, the trend is further illustrated in Fig. 8.7 that CB and CF algorithms outperforms others. However, when only one recommendation is generated, the algorithms behave similarly and their differences become more apparent as the number of recommendations grows.

[3]Note that the number of relevant items varies across users, as each profile contains a different number of ratings.

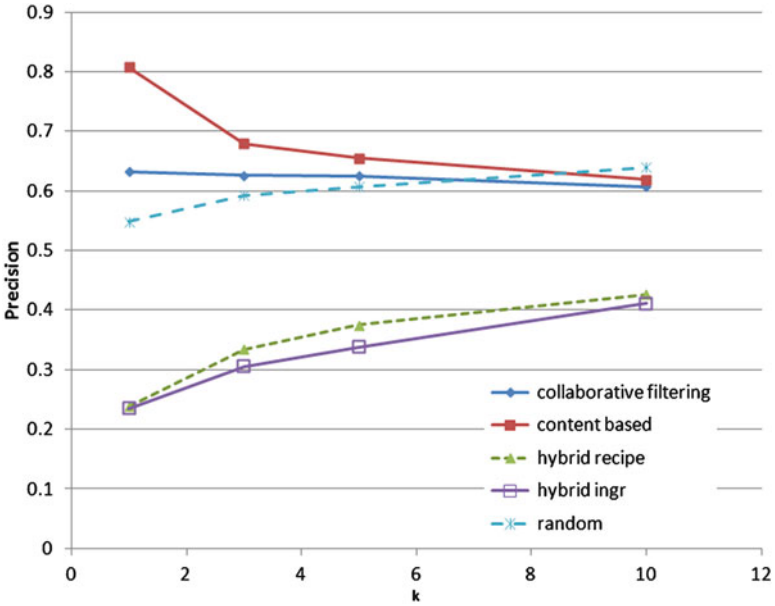

Fig. 8.5 Precision

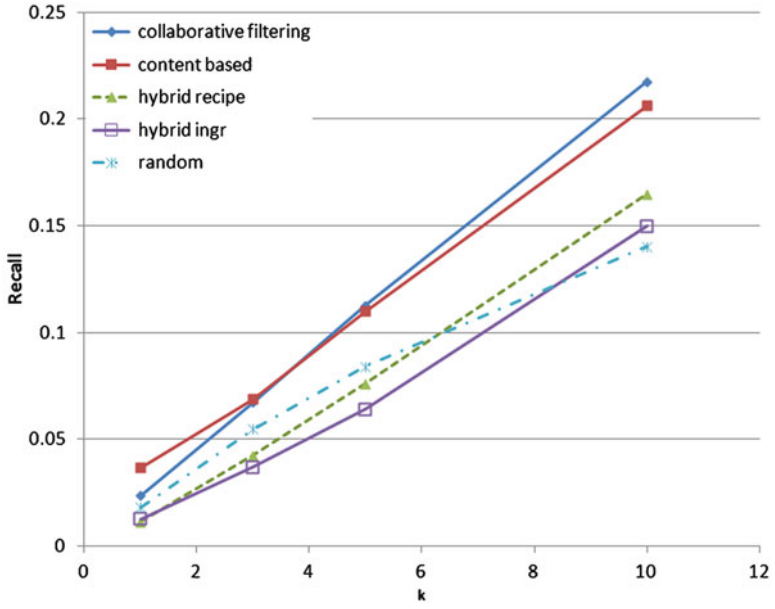

Fig. 8.6 Recall

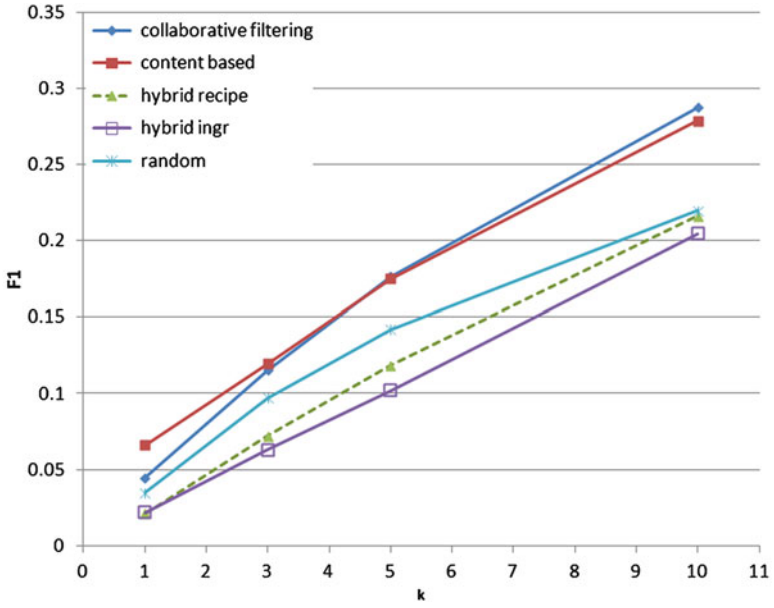

Fig. 8.7 F1 measure

Finally, Fig. 8.8 shows the item space coverage of the algorithms. The lowest coverage of 96% is obtained by the CF algorithm, which exploits the ratings of similar users. The coverage of CF is often impacted by the sparsity of the profiles in the dataset. In some cases, a target user's profile does not contain sufficient or suitable information for accurate neighbourhood formation; in other cases, the sparsity of the neighbours' ratings might result in none of the neighbour profiles containing a rating for the target item. In either scenario, a prediction cannot be generated. Higher coverage above 99% is obtained by all other algorithms.

This offline evaluation has shown the applicability of various personalized algorithms for the prediction of recipe ratings and measured their performance over a number of metrics. In terms of prediction accuracy, the CB and hybrid algorithms outperformed CF. This performance difference was also illustrated when general classification accuracy was assessed. However, in terms of precision, recall, and F1 scores for different sizes of the set of recommended recipes, the CB and CF algorithms clearly outperformed the hybrid methods. In the context of a meal recommendation system, precision is likely to be the most appropriate indicator of applicability, as the predicted scores are unlikely to be shown to users, but rather small sets of meals are likely to be presented to users for inclusion in their plans. Thus, the increased performance of the CB and CF algorithms would make them the most appropriate for the recommender. The coverage results uncovered a potential weakness of the CF algorithm, as its coverage was only around 90%, and slightly prioritised the appropriateness of the CB algorithms.

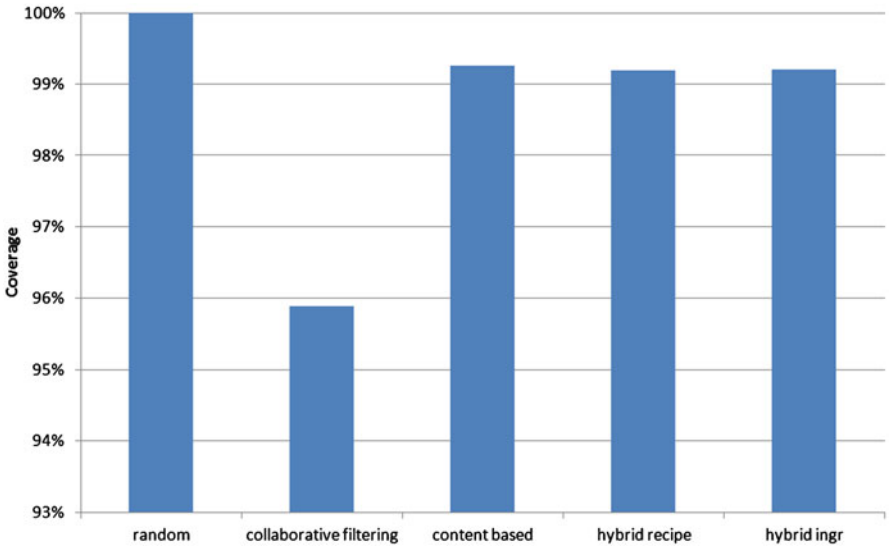

Fig. 8.8 Algorithm coverage

8.3.4 Possible Future Directions

The results of the offline analysis provided information regarding the strengths and weaknesses of each algorithm in the candidate algorithm set, indicated that no algorithm performs badly, and that different algorithms win out for different scenarios. We noted that it was unlikely that predictions scores would be shown to users and that groups of recipes would rather be presented as recommendations, but the dieters may have another opinion on how a meal recommender system should work. We conclude this chapter with a discussion of future studies, which would compliment and strengthen the lessons learned from the conducted offline study. Naturally, these future studies follow the user study and online evaluation paradigms.

8.3.4.1 User Study

There are many avenues for analyses of the suggested meal recommender, which could be achieved through directed user studies. These include interface options, such as plan creation mechanisms, diet compliance visualisations, usage frequencies, the convenience of shopping lists and diet summaries, as well as the straightforward algorithm performance evaluation with real dieters. Understanding the user requirements of planners is also crucial. For example, users may want to add their own recipes to the systems. If so, how should the recommender deal

with these new recipes, their ingredients and the conversion of newly introduced ingredients to their nutritional information? The answers to many of these questions should be understood for effective design of the recommender interface.

A natural next step would be to recruit a cohort of dieters to interact with a prototype meal planner and plan meals for a short period of time and provide their feedback on their experience with the prototype's usability and functionality. The following questions relating to the functionality of the planner and the algorithms applied would be suitable for investigation:

- What is the effect of recommendations on meal planning? Do users create plans faster or with fewer edits?
- What is the accuracy of each algorithm for generating meal recommendations? This would examine the interactions of users in response to recommendations: whether a recommendation is ignored, the recipe is browsed or printed, the recipe is added to the planner, or the consumption of the meal is confirmed.
- What is the effect of recommendations on user satisfaction with the meal planner?

Suitable performance metrics for the algorithms would include classification accuracy metrics, ranking accuracy metrics, as well as the indicators of the time spent planning, dietary compliance, and overall user satisfaction.

8.3.4.2 Online Evaluation

As mentioned previously, offline and usability evaluations can inform researchers about the accuracy and usability of the planner. However, only real users can interact with the system independently and allow researchers to can get a true understanding of the performance of the technology. For example, an online study could ascertain not only if users receive good recommendations, but whether the recommendations are acted upon, i.e., cooked and eaten. In terms of behaviour change and long-term health goals, a longitudinal online study is the only way to investigate the impact of the meal planner on weight loss.

If we consider suitable online evaluations of the meal planner, the following questions relating to the impact of the presence of personalization would be suitable for investigation:

- What is the effect of recommendations on the compliance with the diet? Do personalized suggestions make diet compliance easier?
- What is the effect of recommendations on user satisfaction with the meal planner?
- What is the effect of long-term system usage of personalized recommendations? Do personalized recommendations sustain engagement with the system?

Real dieters, embarking on real diets could be recruited for the purpose of the online evaluation through a live site, and their interactions with the system would be monitored over an extensive period of time. Users could be exposed to various

algorithms or interfaces at different times or assigned one algorithm or interface for the entire duration of the study. The metrics suitable to compare algorithm performance would include ranking and classification accuracy metrics, general success measures, such as uptake, days planned, average diet compliance, and other indicators that could reflect the overall impact of the recommendation component.

8.4 Conclusions

Recommender systems play a key role in assisting user decision making in situations where a large number of options is available. They can also be helpful in assisting users with special needs in planning and supporting their daily routines. In this chapter, we focused on the evaluation techniques of recommender systems and detailed three widely used evaluation paradigms: offline analyses, user studies, and online studies. We provided examples of each paradigm and detailed on specific evaluation metrics suitable for judging various aspects of an algorithm's performance.

We exemplified the use of offline evaluations with a case study of a meal recommender for users with special dietary requirements. The case study compared the performance of collaborative, content-based, and hybrid recommender algorithms with respect to several evaluation metrics and allowed us to derive conclusions regarding the appropriateness of the algorithms for meal recommendations. We also discussed scenarios for other evaluation paradigms, such as a user study and online evaluation, and the research questions that can be successfully addressed by these evaluations.

References

1. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans knowl Data Eng 17: 734–749
2. Bambini R, Cremonesi P, Turrin R (2011) A recommender system for an iptv service provider: a real large-scale production environment. In: Recommender systems handbook. Springer, Boston, pp 299–331
3. Berkovsky S, Freyne J (2010) Group-based recipe recommendations: analysis of data aggregation strategies. In: Amatriain X, Torrens M, Resnick P, Zanker M (eds) RecSys. ACM, New York, pp 111–118
4. Billsus D, Pazzani M (1998) Learning collaborative information filters. In: Proceedings of the fifteenth international conference on machine learning, vol 54. Morgan Kanfinan Publishers Inc. San Francisco, CA, USA, p 48
5. Bollen D, Knijnenburg B, Willemsen M, Graus M (2010) Understanding choice overload in recommender systems. In: Proceedings of the fourth ACM conference on recommender systems. ACM, New York, pp 63–70

6. Burke R (2007) The adaptive web, chap. In: Hybrid web recommender systems. Springer, Berlin/Heidelberg, pp 377–408. URL http://dl.acm.org/citation.cfm?id=1768197.1768211
7. Burke R, Hammond K, Young B (1996) Knowledge-based navigation of complex information spaces. In: Proceedings of the national conference on artificial intelligence, AAAI Press, Oregan USA, pp 462–468
8. Cremonesi P, Garzotto F, Negro S, Papadopoulos A, Turrin R (2011) Comparative evaluation of recommender system quality. In: Proceedings of the 2011 annual conference extended abstracts on human factors in computing systems. ACM, Vancouver, pp 1927–1932
9. Desrosiers C, Karypis G (2011) A comprehensive survey of neighborhood-based recommendation methods. In: Recommender systems handbook. Springer, Boston, pp 107–144
10. Farzan R, Coyle M, Freyne J, Brusilovsky P, Smyth B (2007) Assist: adaptive social support for information space traversal. In: Proceedings of the eighteenth conference on hypertext and hypermedia, HT '07. ACM, New York, pp 199–208. doi:http://doi.acm.org/10.1145/1286240.1286299. URL:http://doi.acm.org/10.1145/1286240.1286299
11. Farzan R, DiMicco J, Millen D, Geyer W, Brownholtz E (2008) Results from deploying a participation incentive mechanism within the enterprise. In: Proceedings of the SIGCHI conference on human factors in computing science (CHI2008), ACM Press, Florence, Italy
12. Freyne J, Berkovsky S (2010) Intelligent food planning: personalized recipe recommendation. In: Proceedings of the 2010 international conference on intelligent user interfaces (IUI 2010), ACM Press, Hong Kong, China, pp 321–324
13. Freyne J, Berkovsky S, Baghaei N, Kimani S, Smith G (2011) Personalized techniques for lifestyle change. In: Proceedings artificial intelligence in medicine, AIME, Bled, Slovenia, 2–6 Jul 2011, pp 139–148
14. Herlocker J, Konstan J, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 230–237
15. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53. doi:http://doi.acm.org/10.1145/963770.963772
16. Koenigstein N, Dror G, Koren Y (2011) Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In: Mobasher B, Burke RD, Jannach D, Adomavicius G (eds) RecSys. ACM, Chicago, pp 165–172
17. Kohavi R, Longbotham R, Sommerfield D, Henne R (2009) Controlled experiments on the web: survey and practical guide. Data Min Knowl Dis 18(1):140–181
18. Konstan J, Miller B, Maltz D, Herlocker J, Gordon L, Riedl J (1997) GroupLens: applying collaborative filtering to Usenet news. Commun ACM 40(3):87
19. Koren Y, Bell R (2011) Advances in collaborative filtering. In: Recommender systems handbook. Springer, Boston, pp 145–186
20. Lops P, Gemmis M, Semeraro G (2011) Content-based recommender systems: state of the art and trends. In: Recommender systems handbook. Springer, Boston, pp 73–105
21. Masthoff J (2004) Group modeling: selecting a sequence of television items to suit a group of viewers. User Model User-Adap Interact 14(1):37–85
22. McJones P (1997) Eachmovie collaborative filtering dataset, DEC systems research center. http://www.research.compaq.com/src/eachmovie/
23. McNee S, Albert I, Cosley D, Gopalkrishnan P, Lam S, Rashid A, Konstan J, Riedl J (2002) On the recommending of citations for research papers. In: Proceedings of the 2002 ACM conference on computer supported cooperative work. ACM, New York, pp 116–125
24. Noakes M, Clifton P (2005) The CSIRO total wellbeing diet book. Penguin Group, Australia
25. Noakes M, Clifton P (2006) The CSIRO total wellbeing diet book 2. Penguin Group, Australia
26. van Pinxteren Y, Geleijnse G, Kamsteeg P (2011) Deriving a recipe similarity measure for recommending healthful meals. In: Proceedings of the 2011 international conference on intelligent user interfaces, IUI 2011, ACM Press, Palo Alto, CA, USA, pp 105–114

27. Pu P, Chen L (2006) Trust building with explanation interfaces. In: Proceedings of the 11th international conference on intelligent user interfaces, IUI '06. ACM, New York, pp 93–100. doi:http://doi.acm.org/10.1145/1111449.1111475. URL:http://doi.acm.org/10.1145/1111449.1111475
28. Quinlan J (1992) Learning with continuous classes. In: Proceedings of the 5th Australian joint conference on artificial intelligence, Citeseer, pp 343–348
29. Rashid A, Karypis G, Riedl J (2008) Learning preferences of new users in recommender systems: an information theoretic approach. ACM SIGKDD Explor Newslett 10(2):90–100
30. Rashid AM, Albert I, Cosley D, Lam SK, McNee SM, Konstan JA, Riedl J (2001) Getting to know you: learning new user preferences in recommender systems. In: Proceedings of the 7th international conference on intelligent user interfaces, IUI '02. ACM, New York, pp 127–134. doi:http://doi.acm.org/10.1145/502716.502737. URL:http://doi.acm.org/10.1145/502716.502737
31. Ricci F, Rokach L, Shapira B, Kantor P (2010) Recommender systems handbook. Springer, New York
32. Said A, Berkovsky S, De Luca EW (2010) Putting things in context: challenge on context-aware movie recommendation. In: Proceedings of the workshop on context-aware movie recommendation, CAMRa '10. ACM, New York, pp 2–6. doi:http://doi.acm.org/10.1145/1869652.1869665. URL:http://doi.acm.org/10.1145/1869652.1869665
33. Shani G, Gunawardana A (2009) Evaluating recommender systems. Recommender systems handbook. Springer, Boston, pp 257–297
34. Swearingen K, Sinha R (2001) Beyond algorithms: an hci perspective on recommender systems. In: ACM SIGIR 2001 workshop on recommender systems, Citeseer
35. Wang Y, Witten I (1996) Induction of model trees for predicting continuous classes. Working paper series ISSN:1170-487X